



Πτυχιακή Εργασία

Σύστημα ανίχνευσης απότομων αλλαγών στις συνθήκες εσωτερικών χώρων χρησιμοποιώντας ένα δίκτυο αισθητήρων

Ανδριάνα Κουτσούκου, Δήμητρα Μαρκουτσά

Επιβλέπων Καθηγητής: κος Δημήτριος
Αμπελιώτης

Περίληψη

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στους αισθητήρες και δίνονται ορισμοί για βασικές έννοιες που χρησιμοποιούνται στην παρούσα εργασία. Περιγράφονται οι κόμβοι αισθητήρων αλλά και η δημιουργία ασύρματου δικτύου αισθητήρων. Αναλύεται η δομή και η αρχιτεκτονική τους, στα θέματα τοπολογίας, δρομολόγησης, κλιμάκωσης και διαχείρισης ενέργειας. Γίνεται διαχωρισμός hardware από interface. Παρουσίαση σημαντικότερων μειονεκτημάτων και πλεονεκτημάτων καθώς επίσης και παραδείγματα πρακτικών εφαρμογών.

Στο δεύτερο κεφάλαιο γίνεται παρουσίαση του kit sunspot που θα χρησιμοποιηθεί. Περιγράφονται τα τεχνικά χαρακτηριστικά και οι δυνατότητες του, καθώς και η υλοποίηση wsn μέσω αυτού. Αναλύεται το hardware και software που περιλαμβάνει. Παρουσιάζονται τα εργαλεία ανάπτυξης, το περιβάλλον χειρισμού και λογισμικό προσομοίωσης του φυσικού προβλήματος για βέλτιστη λειτουργία και απόδοση του συστήματος.

Στο τρίτο κεφάλαιο περιγράφεται το συγκεκριμένο φυσικό πρόβλημα του πρέπει να λυθεί με την εφαρμογή wsn. Ορίζεται ο συγκεκριμένος χώρος όπως και οι απαιτήσεις καλής λειτουργίας του συστήματος σύμφωνα με τις απαιτήσεις που έχουν τεθεί. Παρουσιάζεται στη συνέχεια ο τρόπος σχεδιασμού και υλοποίησης τόσο σε φυσικό επίπεδο όσο και σε επίπεδο κώδικα.

Περιεχόμενα

Κεφάλαιο 1ο

1.1 Εισαγωγή	2
1.2 Ιστορική αναδρομή	5
1.3 Εφαρμογές των ασύρματων δικτύων αισθητήρων	7
1.4 Τύποι κόμβων αισθητήρων με βάση την εφαρμογή	10
1.5 Χαρακτηριστικά ασύρματων δικτύων αισθητήρων	11
1.6 Δομή κόμβου αισθητήρων	15
1.7 Τοπολογία και χωροθέτηση ασυρμάτων κόμβων αισθητήρων	18
1.8 Αρχιτεκτονική ασύρματου δικτύου αισθητήρων	19
1.9 Zigbee και IEEE 802.15.4	30
1.10 Software ασύρματου κόμβου αισθητήρων	33
1.11 Ασφάλεια στα ασύρματα δίκτυα αισθητήρων	34
1.12 Υλοποιήσεις ασύρματων κόμβων αισθητήρων	35

Κεφάλαιο 2ο

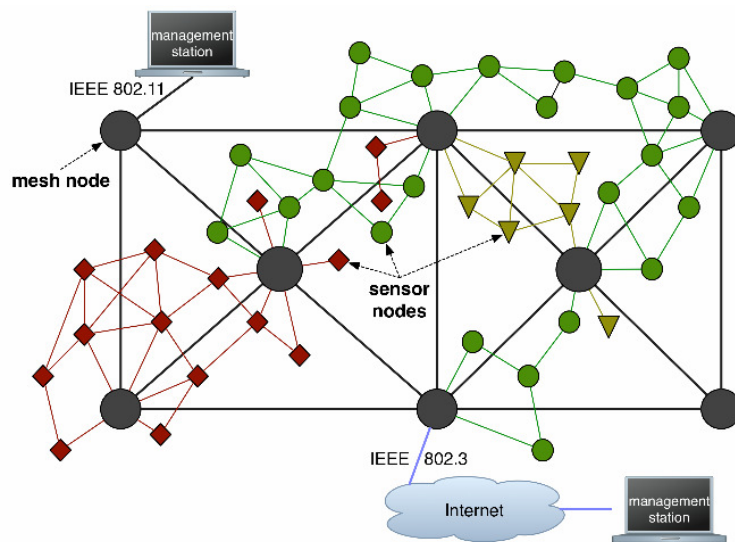
2.1 Διαδίκτυο και Sun	38
2.2 Γλώσσα προγραμματισμού Java	38
2.3 Πλατφόρμα Sun SPOT	40
2.4 Χαρακτηριστικά κόμβου Sun SPOT	42
2.5 Χρήση Sun SPOT	44
2.6 Αγορά Sun SPOT	46

Κεφάλαιο 3ο

3.1 Περιγραφή της εφαρμογής	49
3.2 Εγκατάσταση απαραίτητων εφαρμογών	49
3.3 Χρησιμοποιώντας το Sun SPOT Manager Tool	51
3.4 Χρησιμοποιώντας το Solarium	52
3.5 Κώδικας της εφαρμογής	56
3.6 Παρατηρήσεις	62

Βιβλιογραφία	63
---------------------------	-----------

Κεφάλαιο 1



Ασύρματα δίκτυα αισθητήρων

1.1 Εισαγωγή

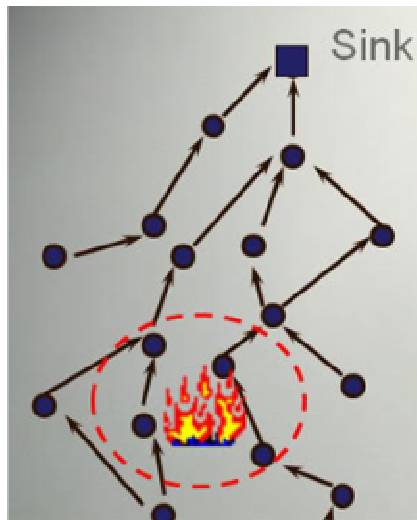
Τα τελευταία χρόνια η αλματώδης πρόοδος στον τομέα των ημιαγωγών οδήγησε στην δημιουργία ολοκληρωμένων κυκλωμάτων με συνεχώς μειούμενο μέγεθος και μικρότερο κόστος παραγωγής. Αποτέλεσμα αυτών η κατασκευή πολύ μικρών υπολογιστικών συστημάτων με κόστος που τα καθιστά αναλώσιμα. Ταυτόχρονες εξελίξεις στον τομέα των ασύρματων επικοινωνιών, σχεδίασης αισθητήρων και αποθήκευσης ενέργειας οδήγησαν στην υλοποίηση των ασύρματων δικτύων αισθητήρων (Wireless Sensor Networks).

Τα δίκτυα αυτά αποτελούνται από ένα μεγάλο αριθμό κόμβων. Κάθε κόμβος είναι μικρός σε μέγεθος και ενσωματώνει διάφορους αισθητήρες για παρακολούθηση φυσικών μεγεθών (θερμοκρασία, κίνηση, πίεση, υγρασία κτλ). Παράλληλα διαθέτει δυνατότητες επεξεργασίας του σήματος και ασύρματες επικοινωνίες τόσο με άλλους κόμβους όσο και με τον σταθμό βάσης (base station ή sink). Κάθε κόμβος για να διατηρεί την αυτονομία του πρέπει να διαθέτει και μια πηγή ενέργειας που συνήθως είναι κάποια μπαταρία.



Τα ασύρματα δίκτυα αισθητήρων έχουν ένα ευρύ φάσμα εφαρμογών το οποίο μεγαλώνει συνεχώς. Όλες όμως οι υλοποιήσεις τους βασίζονται σε μια κοινή αρχή. Τοποθετείται ένας μεγάλος αριθμός κόμβων σε συγκεκριμένη περιοχή, συλλέγουν πληροφορίες τις οποίες μεταδίδουν ασύρματα σε ένα κεντρικό σταθμό. Λόγω του μεγάλου αριθμού κόμβων που συνήθως χρησιμοποιείται και της διασποράς τους στο χώρο, κάθε αισθητήρας δεν

λαμβάνει την ίδια πληροφορία για το ίδιο φαινόμενο. Ο κεντρικός σταθμός είναι υπεύθυνος στο να συντονίσει και να αξιολογήσει τη μέτρηση κάθε κόμβου ώστε να φτάσει στο τελικό συμπέρασμα.



Αν και η ιδέα του δικτύου αισθητήρων είναι σχετικά απλή και οικονομική, η υλοποίηση ενός τέτοιου δικτύου για συγκεκριμένη διεργασία μόνο εύκολη δεν θεωρείται. Υπάρχουν πολλοί παράμετροι που πρέπει να λάβει υπόψη του ο σχεδιαστής τους συστήματος ώστε να επιτευχθεί η μέγιστη αποτελεσματικότητα στο μικρότερο δυνατό κόστος. Βασικό πρόβλημα είναι η τοποθέτηση των κόμβων στο χώρο. Αυτή μπορεί να είναι τυχαία ή να προσδιορίζεται από κάποιον αλγόριθμο. Το είδος του φαινομένου υπό παρακολούθηση και η επιθυμητή ακρίβεια της μέτρησης είναι αυτά που καθορίζουν τον αριθμό και την τοπολογία των κόμβων. Επίσης εγείρονται θέματα ιεραρχίας δικτύου. Η αξιολόγηση της πληροφορίας εξαρτάται από τον τρόπο με τον οποίο συνδέονται μεταξύ τους οι κόμβοι. Άμεση σχέση με αυτό έχουν και οι αλγόριθμοι δρομολόγησης. Η επικοινωνία του κάθε κόμβου με τον τελικό σταθμό βάσης τις περισσότερες φορές δεν είναι απευθείας και πρέπει να επιλεγεί η γρηγορότερη και πιο αξιόπιστη διαδρομή της πληροφορίας. Τα πρωτόκολλα επικοινωνίας εξαρτώνται από το τελευταίο προσθέτοντας νέα προβλήματα στο σχεδιαστή. Εκτός από όλα αυτά τα ζητήματα που βρίσκονται σε θεωρητικό επίπεδο υπάρχουν και θέματα στην υλοποίηση του σχεδιασμού. Το είδος του χώρου που θα τοποθετηθούν οι κόμβοι (κλειστός η ανοικτός χώρος, βιομηχανικό περιβάλλον, υπόγειο κτλ) καθορίζουν την επιλογή συγκεκριμένου προϊόντος επηρεάζοντας το κόστος. Επίσης υπολογίζεται αν πρόκειται για δημιουργία νέου συστήματος ή επέκταση υπάρχοντος καθώς και η συνεργασία με άλλα

υπολογιστικά συστήματα εισάγοντας θέματα συμβατότητας.

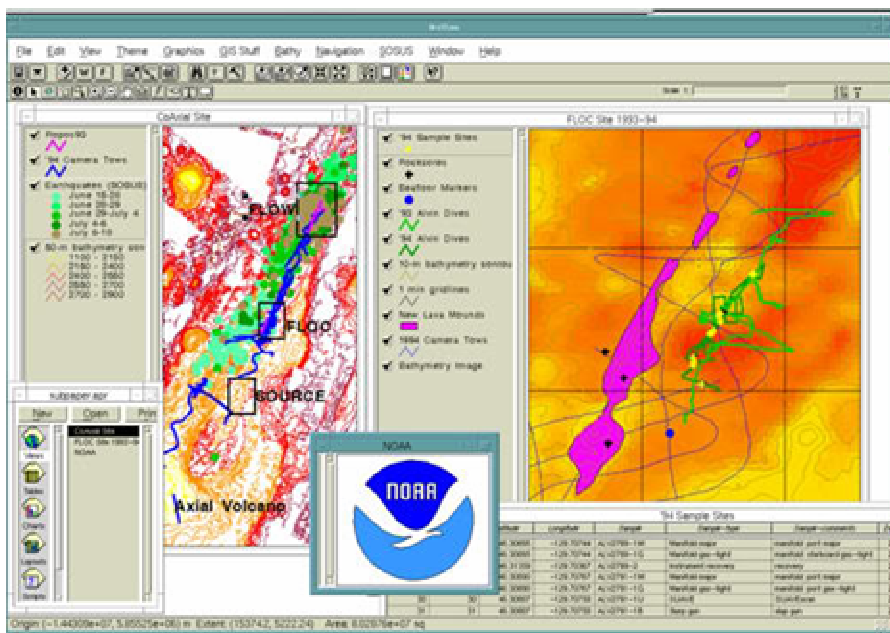
Τα οφέλη των ασύρματων δικτύων αισθητήρων ξεπερνούν κατά πολύ τα όποια σχεδιαστικά προβλήματα ή δυσκολίες. Το μεγάλο φάσμα εφαρμογών και το μικρό τους κόστος είναι οι βασικοί παράγοντες που οδήγησαν στην ευρεία εξάπλωση τους. Η ενσωμάτωση πολλών αισθητήρων σε έναν κόμβο γενικής χρήσης τους δίνει τη δυνατότητα επέκτασης του αρχικού σχεδιασμού του δικτύου αλλά και την προσαρμογή όλου του συστήματος στις εκάστοτε ανάγκες. Το δίκτυο που δημιουργήθηκε για να παρακολουθεί κάποιο συγκεκριμένο φυσικό φαινόμενο έχει τη δυνατότητα χωρίς αλλαγή στο υλικό να παρακολουθεί στο μέλλον και κάποιο άλλο φυσικό μέγεθος αν οι ανάγκες το αιτήσουν. Παρέχεται στο σχεδιαστή ελευθερία κινήσεων και ενσωμάτωση πολλών λειτουργιών σε ένα σύστημα. Ένας άλλος σημαντικός λόγος που οφείλουν τα WSNs την υψηλή δημιουργικότητά τους είναι η ευκολία τοποθέτησης και εγκατάστασης. Η ασύρματη τεχνολογία και η αυτονομία ενέργειας κάθε κόμβου απελευθερώνει από τα δεσμά των καλωδίων. Δεν απαιτούνται ειδικές ηλεκτρολογικές γνώσεις σε αντίθεση με την εγκατάσταση απλών αισθητήρων. Παρέχεται έτσι η δυνατότητα να ενσωματωθεί το WSN αργότερα και όχι κατά την κατασκευή του χώρου στον οποίο θα τοποθετηθεί. Τέλος δεν απαιτούνται εξειδικευμένες γνώσεις ούτε στον τομέα του αυτοματισμού. Με παλαιότερες τεχνολογίες οι αισθητήρες ήταν ένα μικρό κομμάτι μιας αυτοματοποιημένης διαδικασίας αποτελούμενη από PLC, ηλεκτρονόμους (ρελέ) ή μικρο-ελεγκτές. Τώρα οι κόμβοι παρέχουν την δυνατότητα σύνδεσής τους με άλλες ηλεκτρικές διατάξεις. Μέσω του προγραμματισμού του δικτύου, που γίνεται σε μια διαδεδομένη γλώσσα υψηλού επιπέδου, δημιουργείται η επιθυμητή αυτοματοποιημένη διαδικασία. Έτσι με απλές γνώσεις προγραμματισμού επιτυγχάνεται ένας αυτοματισμός. Συνέπεια των πιο πάνω είναι και η επαναχρησιμοποίηση ενός υπάρχοντος WSN σε άλλο χώρο και με άλλο σκοπό.

Στην αγορά υπάρχει πληθώρα τυποποιημένων προϊόντων για δημιουργία ασύρματων δικτύων αισθητήρων. Καθένα παρέχει διαφορετικές δυνατότητες και διαθέτει τεχνικά χαρακτηριστικά για να ικανοποιηθούν οι ποικίλες ανάγκες των καταναλωτών. Καθένα απευθύνεται σε διαφορετικό κοινό ανάλογα με τον βάθος των γνώσεων που απαιτείται αλλά και το βαθμό εξατομίκευσης που προσφέρουν. Οι κατασκευάστριες εταιρίες παρέχουν υλικοτεχνική υποστήριξη

στους καταναλωτές διευκολύνοντας την όλη διαδικασία.

1.2 Ιστορική αναδρομή

Όπως έχει συμβεί και στο παρελθόν με πλήθος άλλων τεχνολογιών η «γέννηση» και η εξέλιξη των δικτύων αισθητήρων (πρόγονων των WSNs) προήλθε από στρατιωτικά ερευνητικά προγράμματα. Η πρώτη γνωστή εφαρμογή δικτύων αισθητήρων υπήρξε το SOSUS (Sound Surveillance System, Σύστημα Ηχητικής Παρακολούθησης). Το παραπάνω δίκτυο χρησιμοποιήθηκε στις αρχές της δεκαετίας του 1950, κατά τη διάρκεια του ψυχρού πολέμου, για την ανίχνευση και τον εντοπισμό Σοβιετικών υποβρυχίων με τη βοήθεια ειδικών ακουστικών αισθητήρων (υδρόφωνα). Το SOSUS παραμένει ακόμη σε λειτουργία, για ειρηνικούς σκοπούς και χρησιμοποιείται κυρίως για την παρακολούθηση διάφορων φαινομένων όπως σεισμικές δραστηριότητες, παρακολούθηση φαλαινών κλπ.



Το επόμενο δίκτυο αισθητήρων αναπτύχθηκε επίσης για στρατιωτική χρήση. Γύρω στο 1980 το πρόγραμμα Distributed Sensor Networks (DSN, Κατανεμημένο Δίκτυο Αισθητήρων) ξεκίνησε από την Defense Advanced Research Projects Agency (DARPA). Ο πρόγονος της DARPA ήταν η Advanced Research Projects Agency (ARPA) που είχε δημιουργήσει παλαιότερα (1969) το δίκτυο



ARPANET, το οποίο υπήρξε ο πρόγονος του σημερινού Διαδικτύου (Internet).

Η δυνατότητα να επεκταθεί το ARPANET στα δίκτυα αισθητήρων απασχόλησε σε μεγάλο βαθμό τον R. Kahn (συν-εφευρέτη του πρωτοκόλλου TCP/IP και με καθοριστικό ρόλο στη δημιουργία και ανάπτυξη του σημερινού διαδικτύου-internet), ταυτόχρονα με την έρευνα σε εμπειρικές τεχνικές ανάλυσης ακουστικού σήματος. Μια από τις πρώτες εφαρμογές που αναπτύχθηκε από το πανεπιστήμιο του Berkley υπό τη χρηματοδότηση του DARPA με υλοποίηση ασύρματου δικτύου αισθητήρων μεγάλης κλίμακας και ονομάστηκε Smart Dust (έξυπνη σκόνη).

Στόχος του προγράμματος αυτού ήταν η δημιουργία μιας αυτόνομης πλατφόρμας σε χιλιομετρική κλίμακα αισθητήρων και επικοινωνιών για μαζικά διανεμημένα δίκτυα αισθητήρων. Το Smart Dust προοριζόταν αρχικά για την εξαποστάσεως παρακολούθηση εχθρικών στρατευμάτων από το στρατό μέσω χιλιάδων ασύρματων μικροαισθητήρων (motes) διασκορπισμένων στο πεδίο της μάχης.



Εκτός από τις στρατιωτικές εφαρμογές το Smart Dust βρήκε πληθώρα εφαρμογών όπως για παρακολούθηση των ατμοσφαιρικών και καιρικών συνθηκών. Αξιοσημείωτο είναι η βιοτεχνολογική προσέγγιση τις ιδέας με motes από χημικά συστατικά αντί για ηλεκτρονικά κυκλώματα.

Σταθμό στην εξέλιξη των WSNs είναι το μικρό, ανοικτού κώδικα, ενεργειακά αυτόνομου λειτουργικού συστήματος γνωστού ως Tiny Micro threading Operating System (TinyOS) που αναπτύχθηκε στα εργαστήρια του πανεπιστημίου του Berkley. Το λειτουργικό αυτό προσφέρει το βασικό πλαίσιο και αναπτυξιακό περιβάλλον για τα WSNs και λειτουργεί σε συνάρτηση με τη ισχύ το μέγεθος και το κόστος. Το TinyOS ελέγχει τόσο τον εξοπλισμό όσο και το δίκτυο, κάνοντας ταυτόχρονα μετρήσεις ,αποφάσεις δρομολογήσεις και έλεγχο και εξοικονόμηση ενέργειας

Σήμερα, τα ασύρματα δίκτυα αισθητήρων κερδίζουν συνεχώς έδαφος. Αντικαθιστούν παραδοσιακούς αισθητήρες δίνοντας νέα προσέγγιση στην δημιουργία αυτοματοποιημένων διαδικασιών. Χρησιμοποιούνται ακόμα και σε

περιπτώσεις που το κόστος και οι συνθήκες καθιστούσαν τη χρήση αισθητήρων απαγορευτικό. Απευθύνονται σε ευρύτερο κοινό με λιγότερες προαπαιτούμενες γνώσεις συμβάλλοντας στη δημιουργικότητα και την τεχνολογική εξέλιξη.

1.3 Εφαρμογές των ασύρματων δικτύων αισθητήρων

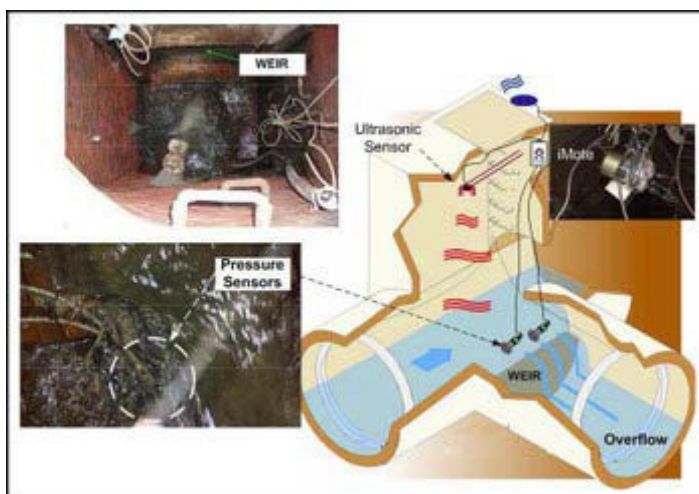
Σήμερα οι εφαρμογές των ασύρματων δικτύων αισθητήρων χωρίζονται σε τρεις κύριες κατηγορίες:

- Εφαρμογές που αφορούν την παρακολούθηση του περιβάλλοντος (εσωτερικού, εξωτερικού, αστικού ή υπαίθριου).
- Παρακολούθηση αντικειμένων (μηχανών και κτιρίων)
- Παρατήρηση της αλληλεπίδρασης και της σχέσης μεταξύ αντικειμένων και περιβάλλοντος.

Μια από τις πρώτες εφαρμογές των δικτύων αυτών ήταν η συλλογή δεδομένων σε περιβάλλοντα που είναι δύσκολο να ελεγχθούν ή που είναι δύσκολο να υφίσταται φυσική παρουσία, όπως τα περιβάλλοντα φυσικών πόρων. Σημαντική επίσης υπήρξε και η χρήση τους για εντοπισμό συμβάντων, όπως σεισμικών δραστηριοτήτων ή κινούμενων αντικειμένων, γεγονός που εισήγαγε την έννοια του εντοπισμού συμβάντος ως μια επιπλέον δυνατότητα στη χρήση των δικτύων αυτών. Μια νέα και σαφώς αναδυόμενη εφαρμογή των δικτύων είναι εκείνη της παρακολούθησης και του εντοπισμού θέσης κάποιου αντικειμένου. Οι εφαρμογές αυτές μπορούν να κατηγοριοποιηθούν περαιτέρω σε στρατιωτικές, περιβαλλοντικές, εφαρμογές στον τομέα της υγείας και εφαρμογές για έξυπνα σπίτια ή άλλες εμπορικές εφαρμογές.

Στις περιβαλλοντικές εφαρμογές περιλαμβάνονται η παρακολούθηση της κίνησης πουλιών, μικρών ζώων και εντόμων, η καταγραφή κρίσιμων περιβαλλοντικών παραμέτρων και συνθηκών που επηρεάζουν το κλίμα της γης, η καταγραφή μετρήσεων από θάλασσα, ξηρά και αέρα, ο εντοπισμός πυρκαγιάς δασών και πλημμυρών, η μελέτη της μόλυνσης και η γεωφυσική και μετεωρολογική έρευνα. Παράδειγμα συγκεκριμένης εφαρμογής είναι η πολυεθνική εταιρία British Petroleum (BP) που συνειδητοποίησε τις τεράστιες προοπτικές των WSN τεχνολογιών και έχει επενδύσει στην ανάπτυξη τους σε μεγάλη κλίμακα. Σε μια πειραματική εφαρμογή μετρούσαν κατά τη διάρκεια

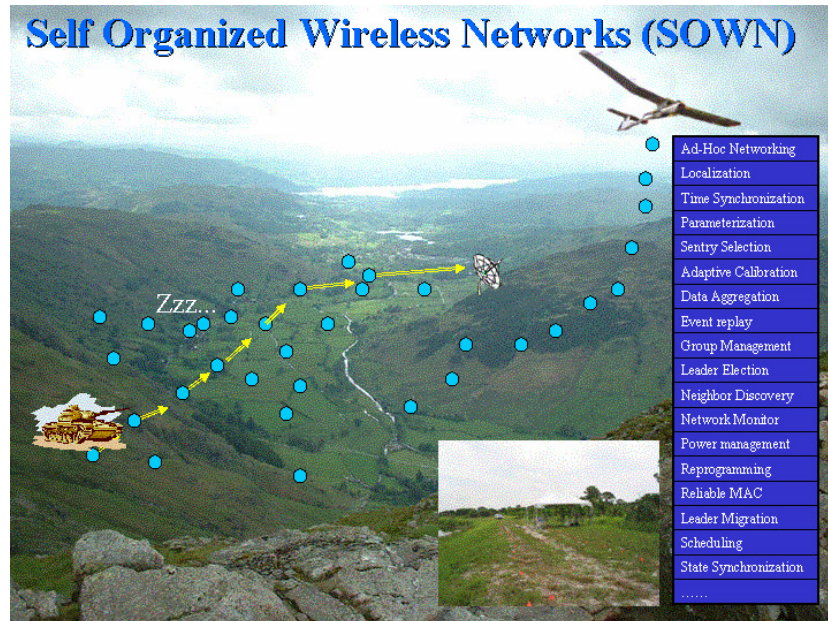
των γεωτρήσεων τους τις μη φυσιολογικές δονήσεις και προειδοποιούν τους μηχανικούς για πιθανή επερχόμενη βλάβη του εξοπλισμού BP που στοχεύει στη χρήση των WSNs για την εξ' αποστάσεως παρακολούθηση του επιπέδου πληρότητας των δεξαμενών υγραερίου. Με τη χρήση υπερηχητικού αισθητήρα στον πάτο της δεξαμενής μετράτε η πληρότητα και εν συνεχεία εκπέμπετε μέσω δορυφόρου χαμηλής τροχιάς σε ένα σταθμό βάσης με αποτέλεσμα να ενημερώνονται οι πελάτες πριν να τελειώσουν τα αποθέματα τους. Το να επιτύχεις τέτοιου είδους κάλυψη με ενσύρματα μέσα δεν θα ήταν απλά μη συμφέρον οικονομικά και δύσκολο να υλοποιηθεί αλλά ανέφικτο



Στις εφαρμογές για οικιακή χρήση τα ασύρματα δίκτυα αισθητήρων συμβάλλουν στην προώθηση των οικιακών αυτοματισμών, στην υλοποίηση «έξυπνων» σπιτιών με περιβάλλοντα που προσαρμόζονται ανάλογα με τις εξωτερικές συνθήκες ή τις επιλογές του τελικού χρήστη.

Στις στρατιωτικές εφαρμογές τα δίκτυα αισθητήρων μπορούν να αποτελέσουν ενιαίο και αναπόσπαστο τμήμα σε συστήματα διοίκησης, ελέγχου, επικοινωνίας, στενής παρακολούθησης, αναγνώρισης εδάφους και συστήματα στόχευσης. Η δυνατότητά τους να λειτουργούν επαρκώς σε αφιλόξενα και επικίνδυνα περιβάλλοντα, τα καθιστούν ιδανικά για χρήση σε παρακολούθηση φίλιων δυνάμεων, εξοπλισμών και πολεμοφοδίων, επιτήρηση πεδίου μαχών, αναγνώριση εχθρικών δυνάμεων, χρήση ως βοηθήματα για στόχευση καθώς και στον εντοπισμό και ανίχνευση βιολογικής, χημικής ή και πυρηνικής επίθεσης. Τέτοιο παράδειγμα αποτελεί το σύστημα VIGILNET. Πρόκειται για ένα μακράς ζωής πραγματικού χρόνου WSN με σκοπό να επαγρυπνά και να δίνει

στρατιωτικές εντολές ώστε να ενημερώνει για κάποια συμβάντα. Αυτά είναι κυρίως παρουσία ανθρώπων, ανθρώπων με όπλα και μικρά ή μεγάλα οχήματα. Όταν η πληροφορία ληφθεί, αναφέρεται σε ένα μετακινούμενο σταθμό βάσης μέσα σε μια αποδεκτή καθυστέρηση. Το σύστημα χρησιμοποιεί πάνω από 200 κόμβους για να παρέχει καλή εποπτεία με ένα κεντρικό σύστημα διαχείρισης ενέργειας.



Συνέπεια αυτού είναι να πετυχαίνει διάρκεια ζωής 3 με 6 μήνες. Αυτό το σύστημα επίσης ενεργοποιεί και εξωτερικούς αισθητήρες όπως υπέρυθρες κάμερες, μόνο όταν κρίνεται απαραίτητο και έτσι αυξάνεται περαιτέρω η διάρκεια ζωής.

Κάποιες από τις εμπορικές εφαρμογές είναι η χρήση τους για καταγραφή της φυσικής καταπόνησης υλικών, η διαχείριση αποθεμάτων και η παρακολούθηση της ποιότητας προϊόντων παραγωγής, η κατασκευή «έξυπνων» χώρων γραφείου και ο έλεγχος περιβάλλοντος σε κτίρια γραφείων, ο αυτοματισμός και ο έλεγχος της επεξεργασίας σε εργοστασιακές μονάδες, η χρήση τους για εντοπισμό και παρακολούθηση κλοπών αυτοκινήτων, για εντοπισμό και καταγραφή της πορείας οχημάτων καθώς και σε πολλές διαδραστικές εφαρμογές μουσείων και χώρων ψυχαγωγίας.

Τα ασύρματα δίκτυα αισθητήρων βρίσκουν ένα σημαντικό πεδίο εφαρμογών στον τομέα της υγείας. Κάποιες από αυτές εντοπίζονται στη συνεχή παρακολούθηση ασθενών, στη διάγνωση, στη σωστή χορήγηση φαρμάκων σε

νοσοκομεία, στην τηλεπαρακολούθηση ανθρώπινων φυσιολογικών δεδομένων και στην καταγραφή θέσης και κατάστασης ασθενών σε νοσοκομεία. Αξιοσημείωτο είναι το σύστημα ALARMNET, ένα WSN για ιατρικές εφαρμογές, σε μεγάλης κλίμακας εγκαταστάσεις συμπεριλαμβάνοντας ετερογενείς συσκευές, τοποθετημένες είτε πάνω στον ασθενή είτε μέσα στο χώρο που περιβάλλει τον ασθενή. Τα στοιχεία που συλλέγονται επεξεργάζονται, αποθηκεύονται και με τη βοήθεια άλλων αισθητήρων (πίεσης, θερμοκρασίας, μόλυνσης του χώρου ακόμη και του ορόφου που βρίσκεται ο ασθενής) δημιουργούν μια πλήρης εικόνα για την κατάσταση του ασθενή μέσα σε απόλυτα ελεγχόμενο περιβάλλον. Κάποια στοιχεία του δικτύου είναι κινητά όπως τα δίκτυα του σώματος, ενώ άλλα είναι σταθερά.

1.4 Τύποι κόμβων αισθητήρων με βάση την εφαρμογή

Το μεγάλο εύρος αλλά και πλήθος εφαρμογών που μπορούν να χρησιμοποιηθούν τα WSNs συνεπάγεται και κατηγοριοποίηση των κόμβων αισθητήρων με βάση το περιβάλλον χρήσης τους. Για να λειτουργεί ο κάθε κόμβος αξιόπιστα και ικανοποιητικά θα πρέπει να είναι προσαρμοσμένος στις εκάστοτε συνθήκες που επικρατούν στο χώρο.

Για περιβαλλοντικές εφαρμογές μεγάλης κλίμακας (π.χ. δάσος, έρημος κτλ.) χρησιμοποιούνται οι επίγειοι κόμβοι αισθητήρων. Είναι σχετικά φθηνοί και μπορούν να τοποθετηθούν ακόμα και χιλιάδες από αυτούς σε μια συγκεκριμένη περιοχή. Μπορούν να αναπτυχθούν με τυχαίο τρόπο όπως ρίψη από αεροπλάνο ή σε προσχεδιασμένες θέσεις από ανθρώπους ή ρομπότ. Διαθέτουν τη δυνατότητα να ξαναοργανωθούν, να σχηματίσουν αυτόματα ένα δίκτυο και να επικοινωνήσουν μεταξύ τους αλλά και με το σταθμό βάσης. Το μεγάλο τους πρόβλημα είναι η περιορισμένη ισχύς των μπαταριών τους.

Οι υπόγειοι κόμβοι αισθητήρων χρησιμοποιούνται σε βιομηχανικές εφαρμογές (πχ υπόγεια ορυχεία, πετρελαιοπηγές κτλ.) όπου απαιτούνται πιο ισχυροί και ανθεκτικοί SNs. Είναι ακριβότεροι από τους επίγειους αφού μπορούν να διασφαλίσουν αξιόπιστη επικοινωνία παρά τα εμπόδια του υπεδάφους (διαφορετικής πυκνότητας και σύστασης πετρώματα, υπόγεια νερά κτλ.). Η ανάπτυξή τους γίνεται πάντα με γνώμονα τη συγκεκριμένη εφαρμογή και τις εκάστοτε δυσκολίες που παρουσιάζει. Το γεγονός αυτό κάνει τη

γενικευμένη παραγωγή τους αδύνατη. Επίσης η συντήρησή τους είναι ιδιαίτερα δαπανηρή. Τέλος παρουσιάζουν τους ίδιους περιορισμούς με τους επίγειους στον τομέα της ενέργειας μάλιστα σε μεγαλύτερο βαθμό αφού αποκλείεται η συμπληρωματική χρήση ανανεώσιμων πηγών ενέργειας.

Μια τρίτη κατηγορία είναι οι υποβρύχιοι κόμβοι αισθητήρων για εφαρμογές κάτω από το νερό. Λόγω της ανθεκτικότητας τους είναι ακριβότεροι από τις προηγούμενες κατηγορίες. Επειδή τα υποθαλάσσια δίκτυα δεν είναι τόσο πυκνά όσο τα επίγεια αλλά και λόγω της πυκνότητας του νερού βασικό πρόβλημα αυτής της κατηγορίας εφαρμογών είναι η διασφάλιση της επικοινωνίας των κόμβων. Παρουσιάζεται υψηλή καθυστέρηση, εξασθένιση σήματος, θόρυβος και περιορισμένο εύρος ζώνης.

Μια ιδιαίτερη κατηγορία κόμβων αισθητήρων είναι αυτή που χρησιμοποιείται σε εφαρμογές πολυμέσων. Σε χαρακτηριστικά μοιάζει με αυτά των επίγειων ωστόσο φέρει μια προσαρτημένη κάμερα. Απαιτεί ισχυρότερη επεξεργασία, μεγαλύτερη αποθηκευτική ικανότητα δεδομένων και εύρος ζώνης καθώς πρέπει να μεταδίδονται σήματα εικόνας, ήχου και βίντεο προς το σταθμό βάσης. Ο ραδιοπομπός και ο επεξεργαστής αυτού του τύπου αισθητήρα θα πρέπει να διαθέτουν συγκεκριμένα χαρακτηριστικά ώστε να διασφαλίζουν ένα ελάχιστο επίπεδο ποιότητας υπηρεσιών, συμπίεση δεδομένων και κωδικοποίηση.

Όλοι οι παραπάνω τύποι αισθητήρων μπορούν να είναι είτε κινητοί είτε σταθεροί. Η φορητότητα αποτελεί μια επιπλέον πρόκληση στα θέματα της δρομολόγησης και τοπολογίας του δικτύου. Παράλληλα όμως δίνουν μεγαλύτερη ελευθερία κινήσεων και επιλογών.

1.5 Χαρακτηριστικά ασύρματων δικτύων αισθητήρων

1.5.1 Περιορισμένο μέγεθος

Βασικό χαρακτηριστικό των κόμβων αισθητήρων είναι το μικρό τους μέγεθος. Αποτελεί βασικό χαρακτηριστικό για χρησιμοποίησή τους σε πληθώρα εφαρμογών καθώς εξασφαλίζει τη φορητότητά τους και την επαναχρησιμοποίησή τους. Μειώνει το κόστος εγκατάστασης αφού δεν είναι απαραίτητες ειδικές κατασκευές και μετατροπές για την τοποθέτησή τους. Ένας

λειτουργικός κόμβος αισθητήρων περιλαμβάνει ένα σύνολο υποσυστημάτων. Τα κυριότερα αυτών είναι αισθητήρες, επεξεργαστής, αποθηκευτικός χώρος, πομποδέκτης και πηγή ενέργειας. Οι ολοένα αυξανόμενες απαιτήσεις για μεγάλη υπολογιστική ισχύ που οδηγεί σε διεύρυνση του φάσματος εφαρμογών έρχονται σε σύγκρουση με την βασική αρχή για μικρό μέγεθος. Η αλματώδης εξέλιξη στον τομέα των μικροεπεξεργαστών έχει δώσει εντυπωσιακά αποτελέσματα αλλά δυστυχώς περιορίζεται από την απουσία αντίστοιχης προόδου στον τομέα των ηλεκτρικών συσσωρευτών. Ενώ υπάρχουν αρκετές ιδέες για νέα ηλεκτρικά αποθηκευτικά στοιχεία η παραγωγή τους είναι ακριβή και περιορίζονται σε επίπεδο πρωτοτύπων. Η τεχνολογία των μπαταριών αποτελεί τροχοπέδη στην ανάγκη ταχύτερων και πληρέστερων κόμβων αισθητήρων και στην καθολική τους χρήση.

1.5.2 Χαμηλή κατανάλωση ισχύος

Κάθε κόμβος είναι αυτόνομος και δεν συνδέεται με κανένα καλώδιο, ούτε τροφοδοσίας. Απαραίτητο για την λειτουργία του είναι η ενσωμάτωση μιας πηγής ενέργειας. Αυτή είναι τις περισσότερες φορές μια μπαταρία ενώ σε συγκεκριμένες περιπτώσεις που πληρούνται οι προϋποθέσεις χρησιμοποιούνται ανανεώσιμες πηγές ενέργειας (πχ ηλιακή ενέργεια). Και στις δυο περιπτώσεις η παρεχόμενη στον κόμβο ενέργεια είναι περιορισμένη. Συνεπώς καθοριστικής σημασίας είναι σωστή διαχείριση της ισχύος και η ελαχιστοποίηση της κατανάλωσής της. Ελαττώνεται έτσι το κόστος και ο χρόνος μεταξύ των εργασιών συντήρησης όπως και το διάστημα που οι κόμβοι μένουν εκτός λειτουργίας λόγω έλλειψης ενέργειας

Σε μερικές εφαρμογές η αναπλήρωση των ενεργειακών πόρων είναι αδύνατη με αποτέλεσμα ο χρόνος ζωής της μπαταρίας να συνδέεται άμεσα με το χρόνο ζωής του κόμβου και συνεπώς του ίδιου του δικτύου. Αν και η εξοικονόμηση ενέργειας αποτελεί πρωταρχικό στόχο, δεν έρχεται χωρίς κόστος. Συνεπώς πρέπει να βρεθεί η ισορροπία μεταξύ ποιότητας υπηρεσιών και αξιοπιστίας με την ενεργειακή οικονομία.

Η ενέργεια σε έναν κόμβο μοιράζεται σε τρεις λειτουργίες: την ανίχνευση των φυσικών μεγεθών, την επεξεργασία των δεδομένων και την επικοινωνία των κόμβων. Υψηλά επίπεδα θορύβου στο περιβάλλον λειτουργίας δυσχεραίνουν τη συλλογή δεδομένων καταναλώνοντας περισσότερη ενέργεια για σωστές μετρήσεις. Μεγάλος φόρτος εργασίας σε κόμβο αυξάνει την

επεξεργασία των δεδομένων αλλά και το ενεργειακό κόστος. Τέλος η επικοινωνία των κόμβων γίνεται περισσότερο ενεργοβόρα όταν παρουσιάζονται σφάλματα που οφείλονται σε θόρυβο και ανακλάσεις.

Σε επίπεδο λογισμικού οι λύσεις διαχείρισης ενέργειας στοχεύουν στο να ελαχιστοποιούν τις επικοινωνίες από τη στιγμή που η εκπομπή και η λήψη κοστίζουν ενεργειακά, δημιουργώντας sleep/wake up καταστάσεις για κόμβους συγκεκριμένων στοιχείων. Το να ελαχιστοποιείται ο αριθμός των μηνυμάτων δεν είναι εύκολο. Απαιτούνται έξυπνα πρωτόκολλα δρομολόγησης και μετάδοσης ώστε να ελαχιστοποιούνται οι συγκρούσεις και οι αναμεταδόσεις. Με καλή δρομολόγηση, σύντομα μονοπάτια και αποφυγή συμφόρησης ή ελαχιστοποίηση μπορεί να επιτευχθεί και αυτό ελαχιστοποιεί τον αριθμό των μηνυμάτων που εστάλησαν.

Υπάρχουν λύσεις οι οποίες κρατούν σε wake up κατάσταση έναν ελάχιστο αριθμό κόμβων για να παρέχουν μια ελάχιστη απαιτούμενη παρακολούθηση των φυσικών φαινομένων ενώ επιτρέπουν στους άλλους να βρίσκονται σε κατάσταση sleep για οικονομία. Για να εξισορροπήσουμε την κατανάλωση ενέργειας σε όλο τον αριθμό των κόμβων, εναλλάσσονται περιοδικά οι κόμβοι που βρίσκονται σε sleep με αυτούς σε wake up.

Πολλές φορές η δρομολόγηση επιλέγεται με τρόπο τέτοιο ώστε να ελαχιστοποιεί τη συνολική ισχύ μετάδοσης, οπότε ελαχιστοποιείται επίσης και η ενέργεια που καταναλώνεται ανά πακέτο πληροφορίας. Όμως όταν όλη η κίνηση δρομολογείται κατά μήκος της διαδρομής ελάχιστης ενέργειας, οι κόμβοι σε αυτή θα εξαντλήσουν τη μπαταρία τους σύντομα, αχρηστεύοντας και άλλους κόμβους, ακόμη και όταν αυτοί διαθέτουν ενέργεια, λόγω απώλειας συνδεσιμότητάς τους με το δίκτυο. Συνεπώς δεν φτάνει να σκεφτόμαστε κατά το σχεδιασμό τη συνολική εξοικονόμηση ενέργειας δικτύου αλλά και κάθε κόμβου.

1.5.3 Κάλυψη και Επεκτασιμότητα

Αμέσως σημαντικότεροι παράγοντες είναι η κάλυψη και η επεκτασιμότητα. Όπως είναι προφανές, είναι σημαντικό για τον τελικό χρήστη να μπορεί να αναπτύξει δίκτυα τα οποία καλύπτουν μια ευρεία περιοχή παρατήρησης. Στο σημείο αυτό πρέπει να τονίσουμε ότι η κάλυψη του δικτύου δεν ταυτίζεται απαραίτητα με την ακτίνα κάλυψης των επικοινωνιακών

συνδέσεων που χρησιμοποιεί ο κάθε κόμβος. Με τη χρήση multihop τεχνικών είναι εφικτή η επέκταση της κάλυψης αρκετά πιο μακριά από την ακτίνα που επιτρέπει ο χρησιμοποιούμενος πομπός. Η επεκτασιμότητα αποτελεί ένα από τα σημαντικότερα χαρακτηριστικά των δικτύων αισθητήρων. Ανάλογα με την εφαρμογή, ο αριθμός των κόμβων που συμμετέχουν κυμαίνεται από μερικούς αισθητήρες έως μερικές εκατοντάδες και θα πρέπει να υπάρχουν κατάλληλοι μηχανισμοί που να επιτρέπουν την προσθήκη νέων κόμβων χωρίς να διαταράσσεται η λειτουργία του δικτύου.

1.5.4 Κόστος

Δεδομένου ότι σε πολλές περιπτώσεις τα δίκτυα αισθητήρων αποτελούνται από πολλούς κόμβους, το κόστος ενός εκάστου κόμβου παίζει σημαντικό ρόλο στη διαμόρφωση του συνολικού κόστους του δικτύου. Έτσι επιδιώκεται το κόστος του κάθε κόμβου να είναι χαμηλό ώστε το συνολικό κόστος του δικτύου να είναι χαμηλότερο.

1.5.5 Αυτόνομη και προγραμματισμένη λειτουργία

Ο κάθε κόμβος μπορεί να εκτελεί αυτόνομα τις βασικές του λειτουργίες. Συγκεκριμένα ξέρει πώς να κάνει μετρήσεις, ποια φυσικά μεγέθη να μετράει, με τι συχνότητα δειγματοληψίας και πώς να μεταδώσει τις πληροφορίες που συνέλεξε. Επίσης διαθέτει τη δυνατότητα να προγραμματίζεται δυναμικά. Το base station μπορεί να στείλει νέα δεδομένα λειτουργίας για κάθε κόμβο τροποποιώντας τη λειτουργία του.

1.5.6 Ανοχή σε σφάλματα

Σημαντικό χαρακτηριστικό των WSNs είναι η ανοχή σε σφάλματα. Κάποιοι κόμβοι μπορεί να πάψουν να λειτουργούν λόγω βλάβης, λόγω αστοχίας κάποιου υποσυστήματος τους ή λόγω παρεμβολών από κάποιο εξωτερικό αίτιο. Στην περίπτωση αυτή, το σφάλμα λειτουργίας ενός κόμβου ή μιας ομάδας κόμβων δεν θα πρέπει να επηρεάζει τη συνολική λειτουργία του δικτύου. Η ιδιότητα αυτή καθορίζει την αξιοπιστία του συστήματος και ορίζεται ως η ικανότητα διατήρησης των λειτουργιών του δικτύου χωρίς διακοπή, σε περιπτώσεις βλάβης κάποιων κόμβων του δικτύου

1.5.7 Χρόνος Απόκρισης

Τα WSNs έχουν την ικανότητα μέσα σε μερικά λεπτά να έχουν

χαρτογραφήσει το δίκτυό τους και να ξεκινούν την προγραμματισμένη λειτουργία τους. Αυτό εξαρτάται άμεσα από την ταχύτητα αποστολής δεδομένων και το χρόνο απόκρισης. Ο χρόνος απόκρισης είναι για κάποιες εφαρμογές, όπως τα συστήματα έγκαιρης προειδοποίησης και συναγερμού, ο σημαντικότερος παράγοντας σχεδίασης και αξιολόγησης ενός δικτύου. Ωστόσο η ικανότητα του δικτύου να έχει μικρό χρόνο απόκρισης έρχεται σε σύγκρουση με άλλους παράγοντες όπως η μικρή κατανάλωση ισχύος. Η αντίφαση αυτή αίρεται με την ενσωμάτωση στο δίκτυο κόμβων που βρίσκονται σε κατάσταση συνεχούς λειτουργίας. Η λύση αυτή εξασφαλίζει τον επιθυμητό χρόνο αντίδρασης αλλά έχει αρνητική επίπτωση στην ευκολία ανάπτυξης του δικτύου.

1.5.8 Ασφάλεια

Τέλος, τα ασύρματα δίκτυα αισθητήρων πρέπει να είναι ικανά να διατηρούν κρυφή την πληροφορία που συλλέγουν, από μη εξουσιοδοτημένους χρήστες. Συνεπώς, για να μπορέσει να διατηρηθεί η μυστικότητα, το δίκτυο πρέπει να υποστηρίζει μηχανισμούς κρυπτογράφησης και αυθεντικοποίησης. Η χρήση τέτοιων τεχνικών επιδρούν αρνητικά, τόσο στην κατανάλωση ισχύος, όσο και στο διαθέσιμο εύρος ζώνης του δικτύου, ενώ η ενσωμάτωση στα μεταφερόμενα πακέτα επιπλέον bits, τα οποία περιέχουν τις πληροφορίες αυθεντικοποίησης, μειώνουν τον αριθμό των χρήσιμων δεδομένων που μπορούν να μεταφερθούν από έναν κόμβο.

1.6 Δομή κόμβου αισθητήρων

Ένας ασύρματος κόμβος αποτελείται γενικά από τέσσερα βασικά υποσυστήματα: τη μονάδα αισθητήρων, τη μονάδα επεξεργασίας, τη μονάδα επικοινωνίας και τη μονάδα ενέργειας. Επιπρόσθετα υποσυστήματα μπορούν να προστεθούν στον κόμβο ανάλογα με την εφαρμογή. Μερικά από αυτά είναι το σύστημα τοπικής εύρεσης, γεννήτρια ρεύματος και ο κινητήρας.

1.6.1 Μονάδα αισθητήρων

Είναι το κύριο στοιχείο του ασύρματου κόμβου και αυτό που το κάνει να διαφέρει από άλλα ενσωματωμένα συστήματα και δυνατότητες επικοινωνίας. Γενικά περιλαμβάνει αρκετούς αισθητήρες που παρακολουθούν το φυσικό περιβάλλον. Κάθε αισθητήρας παρακολουθεί ένα διαφορετικό φυσικό μέγεθος

όπως θερμοκρασία, υγρασία, επιτάχυνση φωτεινότητα κτλ. Εντός συγκεκριμένου εύρους τιμών, παράγει συγκεκριμένο αναλογικό ηλεκτρικό σήμα ανάλογα με την τιμή του μεγέθους που μετράει. Το σήμα μετατρέπεται σε ψηφιακό μέσω ενός ενσωματωμένου A/D converter και οδηγείται στη μονάδα επεξεργασίας. Η ανίχνευση μπορεί να ρυθμιστεί ώστε να είναι συνεχής, περιοδική ή σποραδική. Στη συνέχεια η μονάδα αισθητήρων μπορεί οποιαδήποτε στιγμή να πραγματοποιήσει μέτρηση. Η περιοδική βασίζεται σε κύκλους λειτουργίας όπου γίνονται μετρήσεις ανά περιοδικά διαστήματα και μετά αδρανοποιείται. Τέλος στη σποραδική μετρήσεις γίνονται όταν συμβαίνει κάποιο γεγονός. Αυτές οι ρυθμίσεις επηρεάζουν την κατανάλωση ενέργειας όπως θα δούμε πιο κάτω.

1.6.2 Μονάδα επεξεργασίας

Αποτελεί τον κύριο ελεγκτή όλου του κόμβου, αφού ελέγχει όλα τα άλλα υποσυστήματα. Αποτελείται από έναν μικροεπεξεργαστή (CPU), προσωρινή μνήμη (RAM) και αποθηκευτικό χώρο δεδομένων (Flash memory). Διαχειρίζεται όλες τις διαδικασίες δηλαδή τη συλλογή δεδομένων, εκτέλεση αλγορίθμων και επικοινωνία με άλλους κόμβους για ασύρματη αποστολή δεδομένων. Υπάρχουν αρκετές οικογένειες αυτού του κομματιού που συμπεριλαμβάνουν microcontrollers, microprocessors και field-programmable gate arrays (FPGAs)

1.6.3 Μονάδα επικοινωνίας

Πραγματοποιεί την επικοινωνία μεταξύ των κόμβων με το σταθμό βάσης. Υπάρχουν τρία αναπτυγμένα σχέδια επικοινωνίας στους αισθητήρες που περιλαμβάνουν οπτική παρακολούθηση (optical communication, laser), υπέρυθρη (infrared) και ραδιοσυχνότητες (RF). Η ραδιοσυχνότητα είναι η πιο εύκολη στη χρήση αλλά χρειάζεται κεραία.

1.6.4 Μονάδα ενέργειας

Όπως είδαμε και σε προηγούμενα κεφάλαια, αποτελεί από τα σημαντικότερα συστήματα. Κάθε υποσύστημα του κόμβου τροφοδοτείται από τη μονάδα ενέργειας. Συνήθως χρησιμοποιούνται μπαταρίες αλλά και άλλες πηγές ενέργειας. Χρησιμοποιούνται μπαταρίες μιας χρήσης αλλά και επαναφορτιζόμενες. Ένα ακόμα ηλεκτροχημικό στοιχείο που κερδίζει έδαφος στο θέμα της τροφοδοσίας είναι ο υπερπυκνωτής. Προτερήματά του είναι η

επεκτασιμότητα και η απόδοση σε συνάρτηση με το βάρος και το μέγεθός του. Διαθέτει επίσης μεγάλη πυκνότητα ισχύος (ενέργεια/χρονικό διάστημα) χρήσιμο χαρακτηριστικό για τους κόμβους αισθητήρων για τις εναλλαγές sleep-wake up. Ο χρόνος φόρτισης είναι μικρός, διαθέτουν χαμηλή αντίσταση και δεν επηρεάζονται από τη θερμοκρασία σε αντίθεση με τις κοινές μπαταρίες. Μειονέκτημά τους είναι η μικρότερη αποθήκευση ενέργειας.

1.6.5 Σύστημα τοπικής εύρεσης

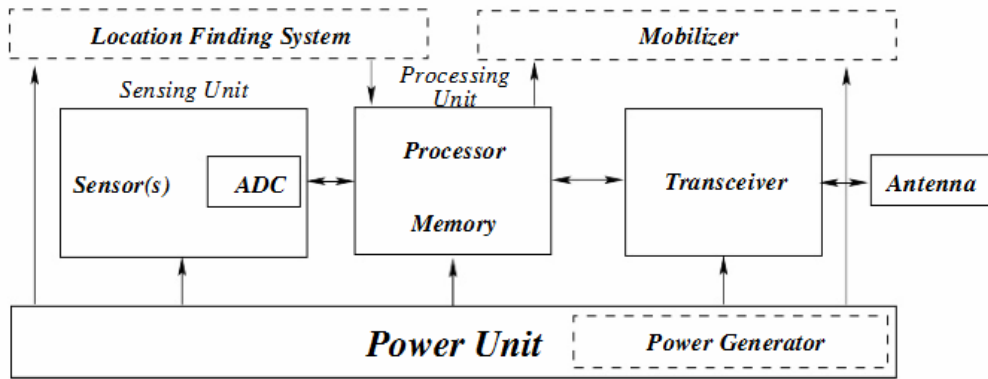
Πολλές από τις εφαρμογές δικτύου αλλά και τεχνικές δρομολόγησης απαιτούν γνώση της ακριβούς φυσικής θέσης του ασύρματου κόμβου. Έτσι είναι συνηθισμένο να περιλαμβάνεται και ένα σύστημα εντοπισμού θέσης. Μπορεί να αποτελείται από ένα κύκλωμα GPS ικανό να καθορίσει την ακριβή θέση του κάθε κόμβου ή να υλοποιείται από ειδικούς αλγόριθμους καθορισμού θέσης μέσα στο δίκτυο. Οι αλγόριθμοι αυτοί κάνουν πολύπλοκους υπολογισμούς χρόνων απόκρισης και καθορίζουν τη σχετική θέση ενός ασύρματου κόμβου μέσα σε συγκεκριμένο δίκτυο σε αντίθεση με το GPS που παρέχει την απόλυτη θέση του κόμβου με υπολογισμό των συντεταγμένων του.

1.6.6 Κινητήρας

Σε μερικές εφαρμογές απαιτείται κινητήρας για τη μετακίνηση του ασύρματου κόμβου. Η δυνατότητα μετακίνησης αυξάνει την απαιτούμενη ενέργεια αφού εκτός από την κίνηση απαιτούνται πολύπλοκοι ενεργοβόροι αλγόριθμοι δρομολόγησης και αποστολής δεδομένων.

1.6.7 Ηλεκτρική γεννήτρια

Σε εφαρμογές που απαιτείται μεγάλη διάρκεια ζωής του κόμβου τοποθετούνται και ηλεκτρικές γεννήτριες. Σε εφαρμογές σε εξωτερικό χώρο και εφόσον η εφαρμογή το επιτρέπει, συνηθίζεται η χρήση φωτοβολταϊκών. Συναντώνται και άλλες τεχνολογίες όπως μετατροπή θερμικής, κινητικής και ενέργεια από δονήσεις σε ηλεκτρική για επαναφόρτιση της μπαταρίας. Οι διαφορετικές υλοποιήσεις οφείλονται στις διαφορετικές απαιτήσεις σε φορτίο και στις διαφορές στις περιοχές εφαρμογής των αισθητήρων.



Η μονάδα επεξεργασίας και επικοινωνίας στις περισσότερες εφαρμογές των κόμβων αισθητήρων απαιτείται να λειτουργούν για κάποια βραχύχρονη περίοδο, η οποία ακολουθείται από έναν κύκλο αδράνειας. Αυτό έχει ως αποτέλεσμα ο κόμβος να καταναλώνει πολύ λίγη ενέργεια για το μεγαλύτερο χρονικό μέρος και ενέργειες όπως επεξεργασία, λήψη και μετάδοση δεδομένων, που απαιτούν αρκετή ενέργεια, διαρκούν για μικρό χρονικό διάστημα. Η μέθοδος αυτή είναι γνωστή ως κύκλος ζωής και συντελεί στην επέκταση της διάρκειας ζωής της μπαταρίας. Η τελευταία θα πρέπει να διαθέτει ικανή πυκνότητα ισχύος ώστε να ανταπεξέλθει στις απότομες αυξήσεις τάσης από την αδράνεια στην ενεργή φάση του κόμβου.

1.7 Τοπολογία και χωροθέτηση ασύρματων κόμβων αισθητήρων

Στη δικτύωση υπολογιστών η τοπολογία σημαίνει το πώς συνδέονται και επικοινωνούν οι κόμβοι μεταξύ τους. Η τοπολογία ειδικότερα χωρίζεται σε 2 μέρη τη φυσική και τη λογική. Στη φυσική τοπολογία εννοούμε το φυσικό σχέδιο που περιλαμβάνει τις συσκευές που απαρτίζουν το δίκτυο και την τοποθεσία τους. Η λογική τοπολογία αναφέρεται στο πώς μεταφέρονται τα δεδομένα μέσα στο δίκτυο όπως θα συνέβαινε στο φυσικό σχέδιο. Τοπολογία μπορεί να θεωρηθεί το εικονικό σχέδιο της δομής ενός δικτύου.

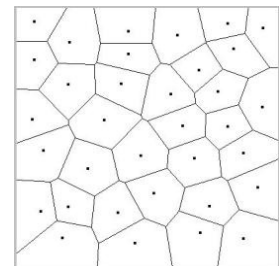
Η τοποθέτηση των ασύρματων κόμβων αισθητήρων σε ένα χώρο για την παρακολούθηση φυσικών μεγεθών μπορεί να είναι σχεδιασμένη ή τυχαία. Όταν το πεδίο εφαρμογής είναι ένας μεγάλος σε έκταση εξωτερικός χώρος με αρκετές εκατοντάδες κόμβους συνήθως η χωροθέτησή τους είναι τυχαία. Σε αυτή την περίπτωση δεν εξασφαλίζεται η σωστή παρακολούθηση του υπό εξέταση φυσικού μεγέθους, πρόβλημα που συνήθως λύνεται με αύξηση του αριθμού των

κόμβων. Ενώ με τον τρόπο αυτό εξασφαλίζεται η αξιοπιστία, αυξάνεται το κόστος. Υπάρχουν συγκεκριμένες τεχνικές για τοποθέτηση των κόμβων εξασφαλίζοντας τη σωστή κάλυψη με τον ελάχιστο αριθμό κόμβων. Αν και τα περισσότερα επίγεια δίκτυα μοντελοποιούνται σε δισδιάστατες δομές, στην πραγματικότητα λειτουργούν στον τρισδιάστατο χώρο. Όμως η τρίτη διάσταση συνήθως είναι πολύ μικρότερη συγκριτικά με τις άλλες δύο οπότε δεν υπάρχουν σημαντικές αποκλίσεις.

Η εμβέλεια ενός αισθητήρα είναι σφαιρική. Επειδή ο διαμελισμός του χώρου σε σφαίρες είναι αδύνατος, χρησιμοποιούνται επικαλυπτόμενα πολυέδρα τα οποία προσεγγίζουν την σφαίρα. Η ακτίνα από το κέντρο του πολυέδρου πρέπει να είναι μικρότερη από την εμβέλεια του αισθητήρα. Σχήματα που χρησιμοποιούνται σε αυτή την τεχνική είναι το τριγωνικό πρίσμα, ο κύβος, το κόλουρο οκτάεδρο κτλ.

Άλλη τεχνική είναι η διάταξη του πλέγματος. Στο δισδιάστατο χώρο υπάρχουν πέντε βασικοί τύποι πλεγμάτων (τετραγωνικό, ρομβικό, εξαγωνικό, ορθογώνιο και παραλληλόγραμμο).

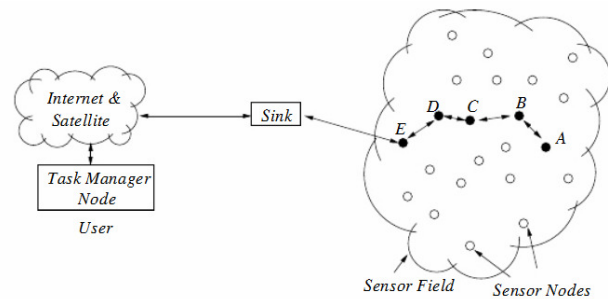
Τέλος χρησιμοποιείται η διαμέριση Voronoi. Σύμφωνα με αυτή για οποιοδήποτε διακριτό σύνολο σημείων S που ανήκει στον ευκλείδειο χώρο, το σύνολο των σημείων που είναι πλησιέστερα σε ένα σημείο c που ανήκει στο S αποτελεί το εσωτερικό ενός κυρτού πολυέδρου που ονομάζεται κυψέλη Voronoi του c .



Οι λογικές τοπολογίες γενικώς καθορίζονται από τα πρωτόκολλα επικοινωνίας που εξετάζονται πιο κάτω. Εάν είναι γνωστή η χωροθέτηση των κόμβων και όχι τυχαία διευκολύνεται η σχεδίαση των πρωτοκόλλων και γίνονται πιο αποδοτικά. Η λογική τοπολογία στα ασύρματα δίκτυα αλλάζει δυναμικά. Η απουσία καλωδίων δίνει τη δυνατότητα με παραπάνω από μία σύνδεση ενός κόμβου με τους γειτονικούς του ανάλογα με τις ανάγκες της εφαρμογής. Επίσης μερικοί κόμβοι έχουν και τη δυνατότητα φυσικής μετακίνησής τους προσφέροντας έτσι επιπλέον ελευθερία.

1.8 Αρχιτεκτονική ασύρματου δικτύου αισθητήρων

Οι κόμβοι αισθητήρων είναι συνήθως διασκορπισμένοι σε έναν χώρο δημιουργώντας ένα πεδίο αισθητήρων όπως φαίνεται στο σχήμα. Κάθε ασύρματος κόμβος έχει την δυνατότητα να συλλέξει και να μεταφέρει τα δεδομένα στον



κεντρικό σταθμό και στον τελικό χρήστη. Κάθε κόμβος δεν εκπέμπει απευθείας στο sink αλλά σε γειτονικούς κόμβους. Συνεπώς υπάρχουν πολλές δυνατές διαδρομές για μεταφορά της πληροφορίας και πρέπει να επιλεγεί η βέλτιστη (δρομολόγηση). Ο τελικός χρήστης μπορεί να επικοινωνεί με τον κεντρικό σταθμό είτε απευθείας είτε μέσω δικτύου. Το σύνολο των επικοινωνιών διέπονται από κανόνες και αρχές λειτουργίας που καθορίζονται από τα πρωτόκολλα επικοινωνίας.

1.8.1 Μοντέλο OSI

Στα πρώτα χρόνια ανάπτυξης δικτύων υπολογιστών κάθε κατασκευαστής δημιουργούσε τις δικές του αρχές λειτουργίας και σύνδεσης, τα πρωτόκολλα επικοινωνίας, για τις δικτυακό του εξοπλισμό και εφαρμογές. Συνέπεια αυτού ήταν συσκευές, δίκτυα και εφαρμογές διαφορετικών κατασκευαστών να μην είναι συμβατά μεταξύ τους. Παρουσιάστηκε γρήγορα η ανάγκη δημιουργίας κοινά αποδεκτών πρωτοκόλλων από όλους. Η πρώτη ολοκληρωμένη προσπάθεια τυποποίησης έγινε από τον οργανισμό ISO (International Standard Organization) στις αρχές τις δεκαετίας του '80 εισάγοντας το μοντέλο OSI (Open System Interconnection) το οποίο τμήμα του ισχύει μέχρι και σήμερα (έχει επικρατήσει στοίβα πρωτοκόλλων TCP/IP).

Το μοντέλο OSI υποδιαιρεί τις λειτουργίες ενός τηλεπικοινωνιακού δικτύου σε μια κατακόρυφη στοίβα από επίπεδα, για το καθένα από τα οποία μπορεί να οριστεί κάποιο πρωτόκολλο σε μία συγκεκριμένη υλοποίηση. Κάθε επίπεδο αξιοποιεί τις λειτουργίες του κατώτερου του στη στοίβα επιπέδου, ενώ στόχος του είναι να παρέχει λειτουργικότητα στο αμέσως ανώτερο επίπεδό του. Μία συγκεκριμένη υλοποίηση του μοντέλου, με καθορισμένα πρωτόκολλα για κάθε επίπεδο, ονομάζεται στοίβα πρωτοκόλλων ή απλά στοίβα. Το κάθε πρωτόκολλο υλοποιείται είτε σε υλικό είτε σε λογισμικό. Συνήθως τα κατώτερα

επίπεδα υλοποιούνται στο υλικό ενώ τα ανώτερα σε λογισμικό. Το βασικό χαρακτηριστικό του είναι η διασύνδεση μεταξύ των επιπέδων, η οποία υπαγορεύει τις προδιαγραφές της αλληλεπίδρασής τους. Αυτό σημαίνει ότι ένα επίπεδο υλοποιημένο με κάποιο συγκεκριμένο πρωτόκολλο μπορεί να συνεργαστεί με το γειτονικό του στη στοίβα επίπεδο, το οποίο υλοποιείται με κάποιο άλλο πρωτόκολλο, υπό την προϋπόθεση ότι οι προδιαγραφές του καθενός έχουν δημοσιευθεί και έχουν γίνει αντιληπτές σωστά.

Μοντέλο OSI			
	Μονάδα δεδομένων	Επίπεδο	Λειτουργία
Software	Data	7. Application	Παρέχεται στις εφαρμογές πρόσβαση στο δίκτυο
		6. Presentation	Αναπαράσταση δεδομένων και κρυπτογράφηση
		5. Session	Έλεγχος του διαλόγου μεταξύ των άκρων της επικοινωνίας
	Segments	4. Transport	Αξιοποίηση επικοινωνία από άκρο σε άκρο
Hardware	Packet	3. Network	Καθορισμός διαδρομών και λογικών διευθύνσεων των κόμβων στα πλαίσια ενός διαδικτύου
	Frame	2. Data link	Φυσική διευθυνσιοδότηση (MAC & LLC)
	Bit	1. Physical	Διαδική μετάδοση σήματος μέσω του φυσικού μέσου

Συνήθως τα επίπεδα είναι αυστηρά διαχωρισμένα μεταξύ τους: αξιοποιούν τις υπηρεσίες του κατώτερου επιπέδου τους και προσφέρουν υπηρεσίες στο ανώτερό τους, αλλά το καθένα δεν παρεμβαίνει στις λειτουργίες του άλλου. Αυτός ο λογικός διαχωρισμός των επιπέδων διευκολύνει πολύ τη μελέτη της συμπεριφοράς των πρωτοκόλλων και επιτρέπει τη σχεδίαση πολύπλοκων και αξιόπιστων στοιβών πρωτοκόλλων. Ορισμένες φορές όμως αυτή η αρχή ανεξαρτησίας των επιπέδων παραβιάζεται, για λόγους βελτιστοποίησης της απόδοσης ή αύξησης της λειτουργικότητας, με πρωτόκολλα διαφορετικών επιπέδων να συγχωνεύονται ή να παρεμβαίνουν το ένα στη λειτουργία του άλλου.

1.8.2 Physical layer

Το φυσικό επίπεδο ορίζει όλες τις ηλεκτρικές και φυσικές προδιαγραφές της επικοινωνίας. Σ' αυτές περιλαμβάνονται οι σχηματισμοί των ακίδων, οι επιτρεπτές τάσεις, οι προδιαγραφές των καλωδίων κλπ. Συσκευές φυσικού επιπέδου είναι οι διανεμητές, τα repeaters, οι κάρτες δικτύου, οι προσαρμοστές διαύλου. Οι κυριότερες λειτουργίες και υπηρεσίες του φυσικού επιπέδου είναι:

- Έναρξη και τερματισμός της ηλεκτρικής σύνδεσης μιας επικοινωνιακής συσκευής.
- Συμμετοχή σε διαδικασίες όπου οι επικοινωνιακές συσκευές εξυπηρετούν αποτελεσματικά πολλούς χρήστες (multiplex). Επιλύονται προβλήματα προτεραιότητας πρόσβασης και ελέγχου ροής δεδομένων.
- Διαμόρφωση και αποδιαμόρφωση των ψηφιακών δεδομένων κατά τη μετάδοση από συσκευή σε συσκευή. Για παράδειγμα, τα ψηφιακά ηλεκτρικά σήματα μπορεί να ταξιδέψουν ως αναλογικά σε χάλκινο καλώδιο, μετά σε οπτική ίνα, μετά να μεταδοθούν από ραδιοζεύξη ή δορυφορικά, να φθάσουν πάλι αναλογικά σε χάλκινο καλώδιο και να γίνουν ψηφιακά στον παραλήπτη.

Στα WSNs το physical layer είναι υπεύθυνο για την επιλογή της συχνότητας εκπομπής, τη δημιουργία του φέροντος σήματος, την ανίχνευση του σήματος, την διαμόρφωση και αποδιαμόρφωση και την κρυπτογράφηση των δεδομένων. Η δημιουργία του φέροντος σήματος και η ανίχνευση του σήματος άπτονται περισσότερο του hardware της μονάδας επικοινωνίας.

1.8 3 Data link layer

Το επίπεδο ζεύξης δεδομένων παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά δεδομένων από μια συσκευή ενός τοπικού δικτύου σε άλλη, αλλά και για την ανίχνευση και διόρθωση σφαλμάτων που συμβαίνουν στο φυσικό επίπεδο. Οι μη ιεραρχημένες διευθύνσεις των συσκευών εδώ είναι οι φυσικές (π.χ. MAC διευθύνσεις), δηλαδή είναι προκαθορισμένες και αποθηκευμένες στις κάρτες δικτύου των επικοινωνούντων κόμβων από το εργοστάσιο. Στα τοπικά δίκτυα της οικογένειας πρωτοκόλλων IEEE 802, και σε κάποια άλλα όπως το FDDI, αυτό το επίπεδο μπορεί να διαιρεθεί σε δύο μικρότερα: α) ένα επίπεδο ελέγχου πρόσβασης στο κοινό μέσο, το υποεπίπεδο MAC (Media Access Control) και β) ένα ανώτερο επίπεδο ελέγχου λογικών

συνδέσεων, το υποεπίπεδο LLC (Logical Link Control) όπου επικρατεί καθολικά το πρωτόκολλο IEEE 802.2 ανεξάρτητα από το υποκείμενο πρωτόκολλο MAC ή φυσικού επιπέδου.

Συγκεκριμένα στα WSNs το επίπεδο αυτό είναι υπεύθυνο για το multiplexing των data streams, την πρόσβαση του μέσου (MAC) και τον έλεγχο σφαλμάτων (error control). Εξασφαλίζει αξιόπιστες συνδέσεις point-to-point και point-to-multipoint.

MAC

Τα πρωτόκολλα ελέγχου πρόσβασης στο μέσο ελέγχουν πότε μπορεί ένας κόμβος να χρησιμοποιήσει το μέσο μετάδοσης για να μεταδώσει ή να λάβει πληροφορία. Στην περίπτωση των WSNs το μέσο είναι ηλεκτρομαγνητικά κύματα που διαδίδονται στο χώρο. Στόχος των πρωτοκόλλων αυτών είναι να μειώσουν τις συγκρούσεις κατά την επικοινωνία των κόμβων εξοικονομώντας ενέργεια και μεγιστοποιώντας τη μεταφορά δεδομένων. Συγκρούσεις συμβαίνουν όταν 2 ή περισσότεροι κόμβοι εκπέμπουν δεδομένα ταυτόχρονα. Όταν χρησιμοποιείται η ίδια συχνότητα τότε τα σήματα παρεμβάλλονται το ένα στο άλλο και τα δεδομένα παραμορφώνονται. Το MAC προβλέπει έλεγχο του μέσου αν είναι απασχολημένο, εάν δεν είναι τότε μεταδίδει, αλλιώς περιμένει και προσπαθεί αργότερα. Όταν ανιχνευτεί σύγκρουση, τηρείται τυχαίο διάστημα αναμονής για κάθε κόμβο και επιχειρείται επανάληψη της αποστολής. Σε περίπτωση νέας σύγκρουσης η αναμονή αυξάνει εκθετικά.

Ένα άλλο ενεργοβόρο ζήτημα που παρατηρείται είναι αυτό των αδρανών κόμβων. Συχνά η μονάδα επικοινωνίας σε έναν κόμβο παραμένει ενεργή επειδή περιμένει λήψη δεδομένων. Αυτό οφείλεται στο γεγονός ότι η επικοινωνία των κόμβων δεν γίνεται συγχρονισμένα. Επίσης συχνό είναι και το φαινόμενο κόμβος να λαμβάνει δεδομένα που δεν προορίζονται γι αυτόν. Αυτά απορρίπτονται αλλά ξοδεύεται ενέργεια. Τα προβλήματα αυτά λύνει το πρωτόκολλο MAC με σβήσιμο του κυκλώματος του δέκτη του κόμβου για μικρό χρονικό διάστημα. Η δυσκολία βρίσκεται στον υπολογισμό του διαστήματος. Όμως ο δέκτης ενός κόμβου πρέπει να είναι σε κατάσταση λειτουργίας μέχρι την εμφάνιση του επόμενου μηνύματος που μπορεί να αναφέρεται σε αυτόν. Συνηθισμένη λύση είναι ο συγχρονισμός των κόμβων όπως η τεχνική TDMA. Πρωτόκολλα MAC προσανατολισμένα σε WSN είναι

τα B-MAC, T-MAC και S-MAC.

Error control

Άλλη μια σημαντική λειτουργία του data link layer είναι ο έλεγχος σφαλμάτων στα μεταδιδόμενα δεδομένα. Δύο σημαντικές μέθοδοι στον τομέα αυτό είναι η Forward Error Correction και Automatic Repeat Request. Η χρήση της τελευταίας είναι περιορισμένη λόγω του μεγάλου κόστους επανεκπομπής και συνεπώς χρησιμοποιείται περισσότερο η FEC.

1.8.4 Network layer

Το επίπεδο δικτύου παρέχει τα λειτουργικά και διαδικαστικά μέσα για τη μεταφορά στοιχειοσειρών δεδομένων μεταβλητού μήκους από μια προέλευση σε έναν προορισμό, μέσα από ένα ή περισσότερα ενδιάμεσα δίκτυα, ενώ διατηρεί την ποιότητα εξυπηρέτησης που απαιτεί το επίπεδο μεταφοράς. Το επίπεδο δικτύου εκτελεί λειτουργίες δρομολόγησης, και αναφέρει σφάλματα σχετικά με την παράδοση των πακέτων. Τα routers λειτουργούν στο επίπεδο αυτό διακινώντας δεδομένα σε διασυνδεδεμένα δίκτυα έκαναν το Διαδίκτυο πραγματικότητα. Υπάρχουν και δικτυακοί διακόπτες που σχετίζονται με τις διευθύνσεις (IP). Εδώ υπάρχει μια λογική οργάνωση και τις τιμές των διευθύνσεων τις καθορίζει ιεραρχικά ο τεχνικός των επικοινωνιών. Το πλέον αναγνωρίσιμο παράδειγμα πρωτοκόλλου δικτύου είναι το Πρωτόκολλο Διαδικτύου (IP).

Στα WSNs η απευθείας επικοινωνία του κόμβου με το κεντρικό σταθμό περιορίζεται από τη μικρή εμβέλεια του πομπού. Απαιτούνται λοιπόν κατάλληλα πρωτόκολλα δρομολόγησης χρησιμοποιώντας τους ενδιάμεσους κόμβους ως αναμεταδότες. Οι υπάρχοντες αλγόριθμοι δρομολόγησης δεν καλύπτουν τις ανάγκες των ασύρματων δικτύων. Προορίζονται για τυπικά δίκτυα με σκοπό τη βελτιστοποίηση της ταχύτητας και αξιοπιστίας ενώ στα WSNs απαιτείται βελτιστοποίηση της κατανάλωσης ενέργειας. Τα πρωτόκολλα δρομολόγησης πρέπει να λαμβάνουν υπόψη τα χαρακτηριστικά των WSNs δηλ, ελάχιστη κατανάλωση ενέργειας, φιλτράρισμα δεδομένων, οι κόμβοι αναμεταδότες λαμβάνουν δεδομένα από άλλους γειτονικούς κόμβους και δεν είναι πάντα δυνατή η διευθυνσιοδότηση. Δεν είναι εύκολο ένα γενικό σύστημα διευθυνσιοδότησης. Αντίστοιχα κάθε κόμβος ξεχωρίζεται από τους γειτονικούς του είτε από την φυσική του θέση (συντεταγμένες ή σχετική θέση στο δίκτυο)

είτε από τις μετρήσεις που παίρνει

Στα WSNs η δρομολόγηση τυπικά ξεκινά με την ανακάλυψη κάποιου «γείτονα». Ο κόμβος στέλνει πακέτα και φτιάχνει τοπικές γειτονικές συνδέσεις. Αυτές οι συνδέσεις περιλαμβάνουν την ελάχιστη πληροφορία από κάθε γείτονα αλλά και την τοποθεσία του. Αυτό σημαίνει ότι οι κόμβοι θα πρέπει να ξέρουν εκ των προτέρων την τοποθεσία του γειτονικού κόμβου. Άλλες πληροφορίες σημαντικές είναι η ενέργεια που μένει, η καθυστέρηση που εισάγει ο κάθε κόμβος αλλά και η εκτίμηση της ποιότητας του σήματος. Στους περισσότερους αλγόριθμους δρομολόγησης, τα μηνύματα κατευθύνονται από μια πηγή και η διεύθυνση του προορισμού βασίζεται σε γεωγραφικές συντεταγμένες και όχι στην ταυτότητα του κάθε κόμβου.

Ένας τυπικός αλγόριθμος που κάνει αυτή τη δουλειά είναι ο Geographic Forwarding (GF). Σε αυτόν κάθε κόμβος γνωρίζει τη θέση του στο δίκτυο. Το μήνυμα το οποίο δρομολογείται περιέχει τη διεύθυνση προορισμού. Φτάνοντας σε κάποιον κόμβο, μπορεί να υπολογίσει ποιος γειτονικός κόμβος είναι πιο κοντά στο προορισμό. Μετά αυτός προωθεί το μήνυμα στην επόμενη δρομολόγηση. Σε παραλλαγές του GF, ο κόμβος μπορεί να υπολογίσει και τις καθυστερήσεις, την αξιοπιστία της σύνδεσης και την ενέργεια που απομένει. Επειδή τα δεδομένα συλλέγονται από πολλούς αισθητήρες παρουσιάζεται συχνά σημαντικός πλεονασμός (πολλαπλή μετάδοση της ίδιας πληροφορίας μέσα στο δίκτυο) και συνεπώς σε άσκοπη κατανάλωση ενέργειας. Η παράμετρος αυτή πρέπει να λαμβάνεται υπόψη από τα πρωτόκολλα δρομολόγησης.

Από πλευράς οργάνωσης τα πρωτόκολλα χωρίζονται σε δύο κατηγορίες, τα επίπεδα και τα ιεραρχικά. Στα επίπεδα κάθε κόμβος έχει τον ίδιο ρόλο και λειτουργία με τους άλλους του δικτύου. Όταν κάποιος από αυτούς πρέπει να στείλει δεδομένα προς τον κεντρικό σταθμό εντοπίζεται η διαδρομή χρησιμοποιώντας τεχνικές υπερχείλισης. Τα επίπεδα πρωτόκολλα είναι αρκετά αποτελεσματικά σε σχετικά μικρά δίκτυα ενώ αποτυγχάνουν σε μεγάλα και πυκνά. Τα ιεραρχικά πρωτόκολλα δρομολόγησης έχουν περισσότερες δυνατότητες επέκτασης και αποθήκευσης ενέργειας. Σε αυτά οι κόμβοι διαδραματίζουν διαφορετικούς ρόλους μέσα στο δίκτυο και ουσιαστικά είναι οργανωμένοι σε σύμπλεγμα. Οι κόμβοι αυτοοργανώνονται σε ομάδες σύμφωνα με τις απαραίτητες προϋποθέσεις ή μετρήσεις. Κάθε ομάδα ή σύμπλεγμα αποτελείται από τον ηγετικό κόμβο και τους κόμβους μέλη. Οι ηγετικοί κόμβοι

μπορούν να οργανωθούν περαιτέρω σε ιεραρχικά επίπεδα.

Γενικά το ιεραρχικό σύστημα προσφέρει περισσότερες δυνατότητες κλιμάκωσης, λιγότερη κατανάλωση ενέργειας και συνεπώς μεγαλύτερη διάρκεια ζωής. Αυτό οφείλεται στο γεγονός ότι μέσα στα συμπλέγματα εκτελείται το μεγαλύτερο μέρος της ανίχνευσης, επεξεργασίας και επικοινωνίας, ενώ πολλές εφαρμογές απαιτούν τη συνολική αξία της μέτρησης ενός χώρου και όχι κάθε κόμβου χωριστά. Σε τέτοιες εφαρμογές η συγκέντρωση δεδομένων από τους ηγετικούς κόμβους βοηθά στην αποσυμφόρηση και εξοικονόμηση ενέργειας.

1.8.5 Transport layer

Το επίπεδο μεταφοράς διεκπεραιώνει τη μεταφορά των δεδομένων από χρήστη σε χρήστη, απαλλάσσοντας έτσι τα ανώτερα επίπεδα από κάθε φροντίδα να προσφέρουν αξιόπιστη μεταφορά δεδομένων από το ένα άκρο της επικοινωνίας στο άλλο. Το επίπεδο μεταφοράς ελέγχει την αξιοπιστία ενός χρησιμοποιούμενου καναλιού με έλεγχο ροής, κατάτμηση και αποτμηματοποίηση, καθώς και έλεγχο σφαλμάτων. Ορισμένα πρωτόκολλα καταγράφουν καταστάσεις και συνδέσεις, οπότε κρατούν λογαριασμό των πακέτων και επανεκπέμπουν αυτά που δεν παρελήφθησαν σωστά. Το συνηθέστερο παράδειγμα πρωτοκόλλου μεταφοράς είναι το TCP. Άλλα πρωτόκολλα μεταφοράς είναι τα UDP κλπ.

Στα WSNs το επίπεδο αυτό είναι ιδιαίτερα σημαντικό όταν το δίκτυο αισθητήρων θα επικοινωνεί με το internet ή άλλα εξωτερικά δίκτυα. Το πρωτόκολλο TCP που χρησιμοποιείται κατά κύριο λόγο στα δίκτυα, δεν ανταποκρίνεται στις ιδιαιτερότητες των WSNs. Στα δίκτυα αισθητήρων η επικοινωνία των κόμβων δεν βασίζεται σε μια καθολική διευθυνσιοδότηση αλλά σχετική εξαρτώμενη από τη θέση του κόμβου ή τις μετρήσεις του. Επίσης κάθε κόμβος έχει περιορισμένη υπολογιστική ισχύ και μνήμη. Συνεπώς δεν μπορεί να λάβει και να αποθηκεύσει μεγάλα δεδομένα όπως οι servers στο internet. Ένα πρωτόκολλο προσανατολισμένο σε αυτές τις ανάγκες είναι το Reliable Multi-Segment Transport (RMST)

1.8.6 Session layer

Το επίπεδο συνόδου ελέγχει τις συνόδους (δηλαδή τις ανταλλαγές δεδομένων) μεταξύ δύο υπολογιστών, του A και του B. Ξεκινά, διαχειρίζεται

και τερματίζει τη σύνδεση μεταξύ μιας τοπικής και μιας απομακρυσμένης εφαρμογής. Αντιμετωπίζει λειτουργίες FDX (full duplex, οι A και B μιλούν ταυτόχρονα από δύο κανάλια) ή HDX (half-duplex, μιλάει ο A και μετά απαντάει ο B από το ένα διαθέσιμο κανάλι), ενώ υποστηρίζει διαδικασίες αποθήκευσης κατάστασης, αναβολής, τερματισμού και επανεκκίνησης. Αυτό το επίπεδο είναι υπεύθυνο για το ομαλό κλείσιμο της συνόδου. Στα ασύρματα δίκτυα αισθητήρων δεν έχει εφαρμογή.

1.8.7 Presentation layer

Το επίπεδο παρουσίασης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών. Στο επίπεδο αυτό τα δεδομένα υφίστανται κρυπτογράφηση, συμπίεση, κωδικοποίηση MIME και όποια άλλη διαμόρφωση απαιτεί η μορφή δεδομένων ή ο σχεδιαστής του πρωτοκόλλου. Παραδείγματα αποτελούν η μετατροπή αρχείων από κώδικα EBCDIC σε κώδικα ASCII και η μετατροπή της δομής των δεδομένων σε μορφή XML ή αντίστροφα (π.χ. από XML σε έγγραφο τύπου DOC). Στα ασύρματα δίκτυα αισθητήρων δεν έχει εφαρμογή.

1.8.8 Application layer

Το επίπεδο εφαρμογών παρέχει στον χρήστη έναν τρόπο να προσπελάσει μέσω μιας εφαρμογής τις πληροφορίες ενός δικτύου. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με την εφαρμογή και, συνεπώς, με το δίκτυο. Στο επίπεδο αυτό γίνεται η διαχείριση των καταναμημένων εφαρμογών, η αποστολή του ηλεκτρονικού ταχυδρομείου κλπ. Παραδείγματα πρωτοκόλλων επιπέδου εφαρμογών αποτελούν τα Telnet, FTP, SMTP και http.

Το application layer υιοθετήθηκε στα WSNs λόγω της μεγάλης του επιτυχίας στο internet. Όμως η μεγάλη κλίμακας υλοποιήσεις των ασύρματων δικτύων αποκάλυψε ότι το ασύρματο κανάλι είχε μεγάλη επίδραση στα πρωτόκολλα ανώτερου επιπέδου. Πιο συγκεκριμένα ο περιορισμός των πόρων των κόμβων αλλά και το γεγονός ότι το δίκτυο είναι προσανατολισμένο στην εκάστοτε εφαρμογή, οδήγησε σε λύσεις που συνέδεαν και απλοποιούσαν τα σαφώς διαχωρισμένα πρωτόκολλα στην στοίβα πρωτοκόλλων.

Συγκεκριμένες λειτουργίες των WSNs βοήθησαν προς αυτή την κατεύθυνση. Κάθε κόμβος διαθέτει δικό του ρολόι χρονισμού για τις εσωτερικές διεργασίες. Κάθε συμβάν που έχει σχέση με τη λειτουργία του κόμβου

αισθητήρων συμπεριλαμβανομένων των μετρήσεων, της επεξεργασίας και της επικοινωνίας σχετίζεται με τον χρονισμό διαδικασιών που ελέγχεται από το εσωτερικό ρολόι. Επίσης το WSN πρέπει να μπορεί να ταξινομήσει σωστά τα συμβάντα από τους κόμβους αισθητήρων ώστε να μοντελοποιεί με ακρίβεια το φυσικό περιβάλλον. Παρουσιάστηκε συνεπώς η ανάγκη ανάπτυξης πρωτοκόλλων συγχρονισμού.

Το πρόβλημα του συγχρονισμού σε ένα δίκτυο αντιστοιχεί στο πρόβλημα εξίσωσης των υπολογιστικών ρολογιών διαφορετικών συσκευών. Ο συγχρονισμός μπορεί να γίνει σε ολόκληρο το δίκτυο ή τοπικά για γειτονικές συσκευές. Η εξίσωση των στιγμιαίων τιμών των ρολογιών όμως δεν είναι αρκετή, αφού στη συνέχεια αυτά αποκλίνουν. Συνεπώς η διαδικασία αυτή πρέπει να επαναλαμβάνεται σε περιοδικά χρονικά διαστήματα, που καθορίζονται από τις απαιτήσεις για ακρίβεια της κάθε εφαρμογής. Το NTP πρωτόκολλο χρησιμοποιείται για το συγχρονισμό ρολογιών. Χαρακτηριστικά πρωτόκολλα για το συγχρονισμό είναι τα RBS, TPSN και FTSP

Σε πολλές εφαρμογές απαιτείται ο ακριβής προσδιορισμός της τοποθεσίας αλλά και του χρόνου που έγινε η κάθε μέτρηση. Ένα συχνό παράδειγμα τέτοιας εφαρμογής είναι η ανίχνευση θέσης και κίνησης συγκεκριμένων αντικειμένων μέσα σε ένα WSN που συνεπάγεται γνώση της ακριβής θέσης τους μέσα στο χρόνο. Για το λόγο αυτό εισήχθησαν στη στοίβα πρωτοκόλλων των ασύρματων δικτύων και χωρικά πρωτόκολλα.

Ο εντοπισμός των κόμβων είναι ένα πρόβλημα του προσδιορισμού της γεωγραφικής τοποθεσίας του κάθε κόμβου μέσα στο σύστημα. Για κάποιους συνδυασμούς απαιτήσεων και θεμάτων το πρόβλημα λύνεται εύκολα. Αν το κόστος δεν είναι παράγοντας σημαντικός και η ακρίβεια μερικών μέτρων είναι δεκτή, τότε για εξωτερικά συστήματα το σύστημα GPS είναι μια απλή απάντηση. Περισσότερες, από τις λύσεις για εντοπισμό σε ένα WSN είναι είτε βασισμένο στην εμβέλεια ή στην ελεύθερη εμβέλεια. Η πρώτη χρησιμοποιεί διάφορες τεχνικές για να καθορίσει πρώτον τις αποστάσεις μεταξύ των κόμβων και έπειτα να υπολογίσει την τοποθεσία χρησιμοποιώντας κάποιες γεωμετρικές αρχές. Για τον προσδιορισμό των αποστάσεων, επιπλέον hardware συνήθως απαιτείται παραδείγματος χάρη, hardware για τον εντοπισμό της χρονικής διαφοράς της άφιξης του ήχου και των ράδιο-κυμάτων. Αυτή η διαφορά μπορεί να μετατραπεί σε μέτρηση της απόστασης. Στη δεύτερη οι αποστάσεις δεν

προσδιορίζονται ευθέως, αλλά μετρείται ο αριθμός των μεταπηδήσεων που πραγματοποιούνται. Αν και οι μετρήσεις των μεταπηδήσεων είναι καθορισμένες, οι αποστάσεις μεταξύ των κόμβων είναι εκτιμημένες χρησιμοποιώντας μια μέση απόσταση ανά μεταπήδηση, και τότε οι γεωμετρικές αρχές εφαρμόζονται για τον υπολογισμό της τοποθεσίας. Αυτή η τεχνική δεν είναι τόσο αποδοτική όσο η πρώτη που αναφέραμε και συχνά απαιτεί περισσότερα μηνύματα. Ωστόσο, δεν απαιτούν επιπλέον hardware για κάθε κόμβο. Δυο πρόσφατες και ενδιαφέρουσες λύσεις είναι το Spotlight και το Radio Interferometric Geolocation. Και οι δυο τεχνικές προσφέρουν υψηλή ακρίβεια σε εμβέλεις της τάξης των cm.

Επιπλέον απαιτούνται εξειδικευμένες τεχνικές διαχείρισης τοπολογίας στο δίκτυο για τη διατήρηση της σύνδεσης και κάλυψης των WSNs. Οι αλγόριθμοι αυτοί προσφέρουν αποτελεσματικές μεθόδους ανάπτυξης δικτύου που συντελούν στην αύξηση του χρόνου ζωής και της απόδοσης. Τα πρωτόκολλα αυτά καθορίζουν τα επίπεδα ισχύος του σήματος μετάδοσης και τη διάρκεια των λειτουργιών των κόμβων με σκοπό την ελαχιστοποίηση της κατανάλωσης ενέργειας χωρίς να διακινδυνεύει η συνδεσιμότητα των κόμβων. Η βελτιστοποίηση της απόδοσης και του κόστους ενέργειας οδήγησε στην δημιουργία πρωτοκόλλων εφαρμογών για ομαδοποίηση και clustering.

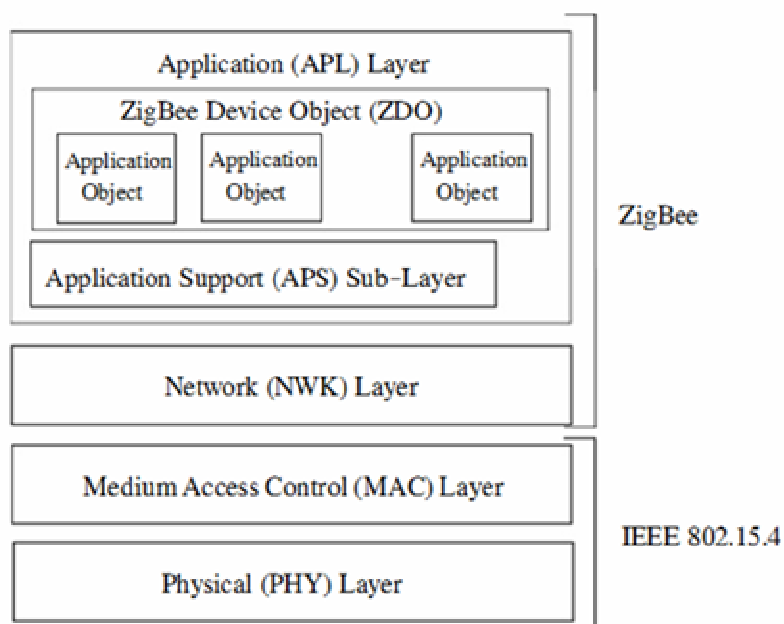
Η ομαδοποίηση είναι μία από τις βασικές προσεγγίσεις για το σχεδιασμό ενεργειακά αποδοτικών, ισχυρών και υψηλής επεκτασιμότητας κατανεμημένων δικτύων αισθητήρων. Κάθε ομάδα έχει έναν ηγέτη, που συχνά αναφέρεται ως ο επικεφαλής-ομάδας (EO) ενώ οι υπόλοιποι κόμβοι είναι απλά μέλη της ομάδας. Οι προτεινόμενες τεχνικές ομαδοποίησης ποικίλλουν ανάλογα με την εγκατάσταση των κόμβων (node deployment), τις τεχνικές με τις οποίες αρχικοποιούνται οι ομάδες στις οποίες θα διαχωριστεί το δίκτυο, την επιδιωκόμενη αρχιτεκτονική του δικτύου, τα χαρακτηριστικά των EO κόμβων, το μοντέλο λειτουργίας του δικτύου και τον επιδιωκόμενο στόχο τους. Ο EO μπορεί να εκλέγεται από τους αισθητήρες που αποτελούν την ομάδα (συστάδα) του ή να είναι μια ιδιότητα η οποία αποδίδεται από τους σχεδιαστές του δικτύου. Μπορεί να είναι απλά ένας από τους αισθητήρες ή ένας κόμβος που είναι πλουσιότερος σε πόρους. Η ιδιότητα του μέλους ομάδας μπορεί να είναι σταθερή ή να αλλάζει. Μέσω της συγχώνευσης (datafusion) και της συνάθροισης (aggregation) των δεδομένων που παρέχουν οι αισθητήρες στα κέντρα των ομάδων, το σύνολο των δεδομένων που αποστέλλονται στον τελικό

προορισμό μπορούν να συρρικνωθούν σημαντικά, εξοικονομώντας ενέργεια και πόρους του bandwidth.

Τέλος το πεδίο εφαρμογών των κόμβων αισθητήρων (επίγειο, υπόγειο, υποθαλάσσιο) καθώς και η φορητότητα (κινητοί κόμβοι) οδηγούν σε επιπλέον προβλήματα και ανάγκες στον τομέα του application layer. Με τόσο διαφορετικές συνθήκες περιβάλλοντος αλλά και μεταβλητές θέσεις δυσκολεύει η δημιουργία γενικών προτύπων με καθολική εφαρμογή.

1.9 Zigbee και IEEE 802.15.4

Η ασύρματη μετάδοση των δεδομένων είναι καθοριστική στα WSNs. Η επιτυχία, η απόδοση και το κόστος ενέργειας είναι σύνθεση πολλών παραμέτρων και το τελικό αποτέλεσμα καθορίζεται από τη συνεργασία πολλών διαφορετικών επιπέδων και πρωτοκόλλων. Η μεγάλη ποικιλία των διαφορετικών υλοποιήσεων ασύρματων κόμβων αισθητήρων από αρκετούς κατασκευαστές συνεπάγεται μεγάλες διαφορές στο hardware και συνεπώς στις δυνατότητές τους. Σε μια προσπάθεια άμβλυνσης των ετεροτήτων προτείνονται συγκεκριμένα πρότυπα υλοποίησης πρωτοκόλλων επικοινωνίας ώστε να είναι δυνατή η συμβατότητα και να διευκολύνεται ο σχεδιασμός του δικτύου αλλά και των εφαρμογών.



1.9.1 IEEE 802.15.4

Το Ινστιτούτο Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE) σχεδίασε το πρότυπο 802.15.4 το οποίο και ανακοίνωσε το 2003. Είναι ένα ανοιχτό πρότυπο κατάλληλο ακόμα και για χαμηλής δυναμικότητας κόμβων αισθητήρων. Στοχεύει στην ελάχιστη δυνατή κατανάλωση ενέργειας με ελάχιστο δυνατό κόστος υλοποίησης. Επίσης επειδή σχεδιάστηκε για να λειτουργεί σε περιβάλλοντα με υψηλά επίπεδα θορύβου έχει μεγάλη ανοχή σε παρεμβολές. Αυτό επιτυγχάνεται με τη χρήση Direct Sequence Spread Spectrum σε φυσικό επίπεδο.

Το πρότυπο αυτό καλύπτει τα physical layer και MAC καθώς και τμήμα του data link layer. Στο φυσικό επίπεδο υποστηρίζει τρεις ζώνες συχνοτήτων 2,4GHz (παγκόσμιο) με ταχύτητα 250Kbps, 915MHz (Αμερική) με ταχύτητα 40 Kbps και 868MHz (Ευρώπη) με 20Kbps. Το MAC επίπεδο του 802.15.4 είναι αρκετά απλούστερο σε σχέση με το αντίστοιχο επίπεδο άλλων προτύπων πχ Bluetooth. Μπορεί να χρησιμοποιηθεί σε διάφορες τοπολογίες δικτύου. Το σύνηθες εύρος χρήσης του είναι από 30 – 50m και κάποιες φορές φτάνει και τα 100m. ενώ η ενεργειακή απόδοση του κύκλου ζωής του δικτύου είναι της τάξης του 1% γεγονός το οποίο καταλήγει στη χρήση μιας πολύ χαμηλής μέσης ισχύος. Λόγω της μεγάλης εξοικονόμησης ενέργειας μπορεί να προσφέρει μεγάλη διάρκεια ζωής.

Κάθε κόμβος ενός δικτύου 802.15.4 διαθέτει μια μοναδική 64-bit διεύθυνση (παρόμοια με την MAC διεύθυνση κόμβων στο Ethernet). Αυτή είναι η extended διεύθυνση της συσκευής. Υπάρχει και μια πιο σύντομη των 16-bit η οποία ανατίθεται από τον συντονιστή του δικτύου όταν η κάθε συσκευή ξεκινά τη λειτουργία της μέσα στο δίκτυο. Η extended χρησιμοποιείται για την επικοινωνία μεταξύ γειτονικών κόμβων ενώ η πιο σύντομη για την επικοινωνία κόμβου με τελικό σταθμό. Από τα bit που χρησιμοποιούνται για τη διευθυνσιοδότηση μπορούμε να υπολογίσουμε και το συνολικό αριθμό κόμβων που μπορούν να συνδεθούν. Τα 16bit δίνουν $2^{16}=65536$ διακριτές διευθύνσεις και συνεπώς 65536 διαφορετικούς κόμβους.

Το πρότυπο 802.15.4 ορίζει δύο διαφορετικούς τύπους συσκευών, τους κόμβους πλήρους λειτουργίας και τους κόμβους περιορισμένης λειτουργίας. Η διάκριση αυτή γίνεται τόσο με βάση τη λειτουργία τους όσο και των δυνατοτήτων τους. Οι κόμβοι πλήρους λειτουργίας μπορούν να λειτουργήσουν

με peer-to-peer τρόπο μεταξύ τους. Μπορούν να δρομολογήσουν κίνηση που λαμβάνουν από γειτονικό κόμβο προς οποιονδήποτε άλλο κόμβο. Αντίθετα οι κόμβοι περιορισμένης λειτουργίας μπορούν να επικοινωνήσουν μόνο με κόμβο πλήρους λειτουργίας. Συνεπώς οι κόμβοι περιορισμένης λειτουργίας προωθούν τις μετρήσεις τους στους κόμβους πλήρους λειτουργίας και οι τελευταίοι αναλαμβάνουν τη δρομολόγησή τους μέσα στο δίκτυο με τελικό προορισμό τον κεντρικό σταθμό.

1.9.2 Zigbee

Το πρότυπο Zigbee χρησιμοποιεί το IEEE 802.15.4 για τα κατώτερα επίπεδα του OSI μοντέλου ενώ υλοποιεί μέρος του data link layer και τα ανώτερα επίπεδα μέχρι το application layer. Έχει σχεδιαστεί ώστε να εκτελείται σε 8-bit επεξεργαστές με μικρό μέγεθος μνήμης. Έτσι η στοίβα των πρωτοκόλλων που υλοποιεί είναι μόλις 32Kb για συσκευές πλήρους λειτουργίας και 6Kb για συσκευές περιορισμένης λειτουργίας.

Το Zigbee περιλαμβάνει πρωτόκολλα εγκατάστασης ενός δικτύου σύνδεσης και αποσύνδεσης από αυτό, ανάθεσης διευθύνσεων σε κόμβους, δημιουργία προγράμματος μετάδοσης (transmission scheduling), δρομολόγησης και κρυπτογράφησης. Στα δίκτυα αυτού του προτύπου μπορούμε να έχουμε επιπλέον τοπολογίες σε σχέση με τις τοπολογίες που προσφέρει το 802.15.4 όπως το cluster tree και mesh. Συγκεκριμένα η τοπολογία mesh χρησιμοποιείται συχνά σε εφαρμογές, επειδή παρέχει πολλές πιθανές διαδρομές προς το κέντρο ελέγχου του δικτύου από διαφορετικούς κόμβους πλήρους λειτουργίας. Το δίκτυο συνεπώς καθίσταται πιο ανθεκτικό σε αστοχίες και σφάλματα. Οι κόμβοι πλήρους λειτουργίας έχουν ανοιχτές τις μονάδες επικοινωνίας τους ενώ οι υπόλοιποι κόμβοι τις ανοίγουν μόνο όταν θέλουν να μεταδώσουν ή να λάβουν μηνύματα. Αυτό συνεπάγεται μικρότερη κατανάλωση ενέργειας στους κόμβους μερικής λειτουργίας αλλά αυξημένη στους πλήρους λειτουργίας. Αυτό αντιμετωπίζεται με περιοδική εναλλαγή της κατάστασης των κόμβων από πλήρους λειτουργίας σε μερική και αντίστροφα, εφόσον όλοι οι κόμβοι αισθητήρων έχουν ίδια χαρακτηριστικά. Ένας άλλος τρόπος επίλυσης είναι η σύνδεση επιπλέον πηγών ενέργειας στους κόμβους πλήρους λειτουργίας.

1.10 Software ασύρματου κόμβου αισθητήρων

Τα πρώτα ασύρματα ενσωματωμένα συστήματα αισθητήρων έτρεχαν πάνω σε προσωπικούς υπολογιστές και χρησιμοποιούσαν κυρίως προγράμματα Linux. Όταν η ανάπτυξη αυτών των δικτύων πέρασε από τους μικροεπεξεργαστές (microprocessors) στους μικροελεγκτές (microcontrollers), το Linux είχε πάψει να αποτελεί την κατάλληλη επιλογή. Οι εφαρμογές των συστημάτων της εποχής εκείνης αναπτύσσονταν κυρίως σε τυπική γλώσσα C ή κατευθείαν σε γλώσσα assembly. Ο προγραμματισμός όμως σε αυτή τη γλώσσα είναι δύσκολο να αναλυθεί και επίσης μπορεί εύκολα να καταλήξει εκτός ελέγχου όταν η πολυπλοκότητα της εφαρμογής αυξηθεί. Σε κλιμακωτά συστήματα το πρόβλημα μπορεί να επιληφθεί με αντικειμενοστραφή προγραμματισμό, ο οποίος καθιστά ευκολότερο το διαχωρισμό πολύπλοκων προγραμμάτων σε ανεξάρτητα, ευκολοσύνθετα στοιχεία. Αλλά ο προγραμματισμός με προσανατολισμό στο αντικείμενο απαιτεί δυναμική παραχώρηση μνήμης και τείνει να απαιτεί περισσότερους προγραμματιστικούς πόρους, κάτι το οποίο τον κρίνει ακατάλληλο για ενσωματωμένα συστήματα (embedded systems).

Η γλώσσα NesC, η οποία αναπτύχθηκε από ερευνητές του Πανεπιστημίου UC Berkeley, είναι κατάλληλα σχεδιασμένη για ενσωματωμένα συστήματα δικτύων και υποστηρίζει ένα προγραμματιστικό μοντέλο που ενσωματώνει αντιδραστικότητα με το περιβάλλον, ταυτοχρονισμό και δυνατότητα επικοινωνίας. Ένα βασικός άξονας επικέντρωσης της NesC είναι ο ολιστικός σχεδιασμός συστημάτων. Οι εφαρμογές των μονάδων-αισθητήρων (motes) είναι βαθιά συνδεδεμένες στο υλικό και κάθε μονάδα τρέχει μια εφαρμογή κάθε φορά.

Επειδή η γλώσσα NesC έχει αποδειχθεί ότι είναι αποτελεσματική στην περίπτωση ανάπτυξης εφαρμογών για ασύρματα δίκτυα αισθητήρων, χρησιμοποιείται σαν την προγραμματιστική γλώσσα για το λειτουργικό σύστημα TinyOS, ένα μικρό λειτουργικό σύστημα για ασύρματα δίκτυα αισθητήρων που έχει υιοθετηθεί από ένα μεγάλο πλήθος ερευνητικών ομάδων σε όλο τον κόσμο. Σε εξέλιξη βρίσκονται έρευνες προς ανάπτυξη στο πρότυπο του TinyOS και άλλων γλωσσών προγραμματισμού αλλά μέχρι στιγμής η NesC είναι η μόνη γλώσσα που μπορεί να χρησιμοποιηθεί για την ανάπτυξη προγραμμάτων στο TinyOS.

Το TinyOS που αναπτύχθηκε και εξελίχθηκε επίσης από το πανεπιστήμιο

του Berkley είναι ένα μικρό σε μέγεθος, ανοικτού κώδικα ενεργειακά οικονομικό στη διαχείριση των αισθητήρων λειτουργικό σύστημα που τρέχει και διαχειρίζεται κάθε κόμβο του δικτύου. Το μέγεθος τέτοιων αρχείων είναι τις τάξης των 200 bytes έτσι το μέγεθος του συνολικού προγράμματος παραμένει το ελάχιστο δυνατό. Το λειτουργικό σύστημα αυτό διαχειρίζεται τόσο το υλικό όσο και το ασύρματο δίκτυο εκτελώντας τις μετρήσεις των αισθητήρων παίρνοντας αποφάσεις δρομολόγησης και ελέγχοντας την κατανάλωση ενέργειας.

Παρά τα πολλά λειτουργικά συστήματα που κατά καιρούς προτάθηκαν για τη διαχείριση των WSNs, το TinyOS επικράτησε με μεγάλη διαφορά κυρίως λόγω του ελεύθερου κώδικα που μπορεί να βρει και να χρησιμοποιήσει ο οποιοσδήποτε και της πληθώρας των υλοποιήσεων που αναπτύχθηκαν. Για την υλοποίηση της επαναχρησιμοποίησης κώδικα το TinyOS εφαρμόζει μια αρχιτεκτονική κατανομής σε επιμέρους στοιχεία. Επιπρόσθετα για τη βελτιστοποίηση της διαχείρισης ενέργειας χρησιμοποιεί ένα μοντέλο εκτέλεσης βασισμένο σε γεγονότα όπου τα γεγονότα οδηγούν τα προγράμματα και οι σχετικοί πόροι αποδεσμεύονται με το πέρας της χρήσης τους. Το TinyOS ακόμα προσφέρει μια σειρά από εφαρμογές και εργαλεία ανάπτυξης όπως το Tossim (προσομοιωτής δικτύων του TinyOS), το deluge και το TinyDB που βοηθούν στην ανάπτυξη και έρευνα των WSNs. Λόγω της αποδοτικής σχεδίασης, της μεγάλης κοινότητας υποστήριξης και του ανοικτού κώδικα το TinyOS έγινε το πλέον διαδεδομένο λειτουργικό σύστημα για τα WSNs.

1.11 Ασφάλεια στα ασύρματα δίκτυα αισθητήρων

Με την ανάπτυξη της τεχνολογίας όλο και περισσότερες συσκευές λειτουργούν ασύρματα. Ο όγκος των δεδομένων που μεταδίδεται μέσω του αέρα έχει αυξηθεί θεαματικά και συνεπώς εγείρονται θέματα ασφάλειας και διασφάλισης προσωπικών δεδομένων από κακόβουλες επιθέσεις και μη εξουσιοδοτημένες προσβάσεις. Ειδικά με τις εφαρμογές των WSNs που υλοποιούνται σε νευραλγικούς τομείς όπως της υγείας, συστημάτων ασφαλείας, στρατιωτικών σκοπών κτλ

Οι επιθέσεις που δέχονται τα ασύρματα δίκτυα χωρίζονται σε παθητικές και ενεργητικές. Οι παθητικές ορίζονται ως η παρακολούθηση και ακρόαση του

καναλιού από μη εξουσιοδοτημένους χρήστες. Στις ενεργητικές δεν έχουμε μόνο παρακολούθηση αλλά και τροποποίηση των πληροφοριών. Αυτές μπορεί να προέρχονται είτε από μέλη του δικτύου, όποτε είναι δύσκολο να εντοπιστούν, είτε από κόμβους εκτός δικτύου που παρακολουθεί τα μεταδιδόμενα πακέτα.

Τα ασύρματα δίκτυα αισθητήρων είναι ευπαθή σε διάφορα είδη επιθέσεων που ως προς τον στόχο τους χωρίζονται σε:

- επιθέσεις διαθεσιμότητας δικτύου (DoS). Ο εισβολέας προσπαθεί να διακόψει, να παρεμποδίσει ή να καταστρέψει το δίκτυο. Επίσης εκτός από τους φυσικούς τρόπους μπορεί να δημιουργήσει εντός του δικτύου πολλαπλά αιτήματα για επικοινωνία με αποτέλεσμα ο συντονιστής να μην μπορεί να ανταπεξέλθει στο μεγάλο όγκο τους. Τεχνικά εύκολη επίθεση αλλά πολύ επικίνδυνη.
- επιθέσεις στο απόρρητο και την αυθεντικότητα. Οι συνηθισμένες τεχνικές κρυπτογράφησης μπορούν να προστατέψουν το απόρρητο και τη γνησιότητα των καναλιών επικοινωνίας από υποκλοπές, αναπαραγωγή πακέτων ή την πλαστογράφηση τους.
- κρυφή επίθεση στην ακεραιότητα των υπηρεσιών. Σε αυτή σκοπός είναι να δεχτεί το δίκτυο μια ψεύτικη τιμή δεδομένων.

Οι τεχνικές ανίχνευσης διακρίνονται σε τεχνικές ανίχνευσης κακής λειτουργίας και ανίχνευσης ανωμαλιών. Η πρώτη κατηγορία ανιχνεύει τα ίχνη από επιθέσεις ενώ η δεύτερη συγκρίνει τη λειτουργία του δικτύου με ένα προκαθορισμένο μοντέλο σωστής λειτουργίας. Οι αλγόριθμοι ανίχνευσης επιθέσεων έχουν να αντιμετωπίσουν και τα εγγενή περιορισμένα τεχνικά χαρακτηριστικά των κόμβων. Συνεπώς υπάρχουν περιορισμοί στην πολυπλοκότητα τόσο λόγω κατανάλωσης ενέργειας όσο και μικρής υπολογιστικής δύναμης. Επίσης περιορισμοί εισάγονται και στον τομέα της επικοινωνίας αφού δεσμεύουν bandwidth αλλά πρέπει να είναι και ανεκτικοί σε σφάλματα.

1.12 Υλοποιήσεις ασύρματων κόμβων αισθητήρων

Εκτός από το Sun Spot της εταιρίας Sun με το οποίο θα ασχοληθούμε στο επόμενο κεφάλαιο υπάρχουν στο εμπόριο και άλλες υλοποιήσεις.

Παρουσιάζονται συνοπτικά οι κυριότερες.

1.12.1 Tmote Sky

Η πλατφόρμα Tmote Sky κατασκευάζεται από την εταιρεία Moteiv. Το tmote-sky είναι μια ασύρματη μονάδα (mote) πολύ χαμηλής κατανάλωσης ισχύος, για χρήση σε δίκτυα αισθητήρων και σε εφαρμογές καταγραφής και παρακολούθησης σχεδιασμένες με σκοπό τόσο την ανεκτικότητα στο θόρυβο όσο και την ευκολία περαιτέρω ανάπτυξης και αξιοποίησης. Αποτελεί εξέλιξη του Telosb και είναι το πιο πρόσφατο προϊόν σε μια σειρά από motes που αναπτύχθηκαν από το Πανεπιστήμιο της California, Berkeley με σκοπό τη χρήση τους σε ασύρματα δίκτυα αισθητήρων. Τα κυριότερα γνωρίσματα και τεχνικά χαρακτηριστικά του είναι

CPU	
Bus Speed	8 MHz
RAM	10 kB
Program Space	48 kB
External Flash	1024KB
Serial Communications	DIO,SPI,I2C,UART
Current (active w/ Radio on)	19 mA
Current (sleep)	5.1 uA
Startup Time	6 us
Voltage	1.8-3.6 V
Radio	
Frequency	2400-2483 MHz
Data rate	250 kbps
Output Power Startup Time	-25 to 0 dBm 580 us
Antenna Type	Inverted-F or SMA Coax
Humidity Sensor	
Humidity Accuracy	3.5% RH
Temperature Accuracy	0.5 °C
Sampling Rate	90 Hz



1.12.2 MicaZ

Αποτελεί την τέταρτη γενιά των MICA motes. Η διαφορά με τους προκατόχους του είναι η ενσωμάτωση του IEEE 802.15.4 στον πομποδέκτη του. Το υπόλοιπο hardware είναι όμοιο με αυτό του Mica2.

CPU	
Bus Speed	8 MHz
RAM	4 Kb
Program Space	128 Kb
Radio	
Frequency	2400-2483 MHz
Data rate	250 kbps

Κεφάλαιο 2



Πλατφόρμα Sun SPOT

2.1 Διαδίκτυο και Sun

Σήμερα, το διαδίκτυο αποτελείται από εκατομμύρια υπολογιστών, όμως η φύση του αλλάζει καθημερινά. Νέοι τύποι συσκευών αρχίζουν να συνδέονται καθημερινά στο διαδίκτυο. Σύντομα αυτοκίνητα, ιατρικές συσκευές ή ακόμη παιχνίδια θα μπορούν να διαχειρίζονται και να ανταλλάσσουν τα δεδομένα μεταξύ τους ανά τον κόσμο. Ο αριθμός αυτών των συσκευών θα ξεπεράσει κατά πολύ αυτόν των υπολογιστών. Ήδη, πάνω από 1.5 δισεκατομμύρια κινητά με λογισμικό Java επικοινωνούν μεταξύ τους σε καθημερινή βάση. Προβλέπεται, κάποια μέρα στο κοντινό μέλλον να επικοινωνούν μεταξύ τους τρισεκατομμύρια τέτοιων συσκευών.

Για χρόνια, η εταιρία SUN πίστευε ότι μια μέρα τα πάντα θα είναι κομμάτι του internet. Αυτό το όραμα, του «internet όλων των πραγμάτων» περιλαμβάνει όχι μόνο υπολογιστές που θα επικοινωνούν μεταξύ τους και θα ανταλλάσσουν δεδομένα, αλλά θα είναι όλα συνδεδεμένα μεταξύ τους όλα θα επικοινωνούν και θα μοιράζονται δεδομένα, συνεχώς. Αυτοκίνητα, δίκυκλα, αστροναύτες, παιχνίδια ακόμη και δέντρα θα μπορούν να συλλέγουν και να παραδίδουν αμφίδρομα πληροφορίες μέσω αυτών των συσκευών (για παράδειγμα τα δέντρα χρησιμοποιώντας τους κατάλληλους αισθητήρες θα μπορούν να εντοπίζουν κλιματικές αλλαγές είτε κάποιες κινήσεις ζώων).

Υπάρχει μεγάλη ποσότητα πληροφορίας για επεξεργασία, διαχείριση και αποθήκευση. Τα τελευταία 24 χρόνια η Sun δημιουργεί τις κατάλληλες υποδομές ώστε να είναι δυνατός ο χειρισμός μεγάλων ποσοτήτων δεδομένων φτιάχνοντας servers, εξειδικευμένες συσκευές δικτύου, συστήματα αποθήκευσης δεδομένων, λογισμικό διαχείρισης δεδομένων κτλ. Με την πλατφόρμα Sun SPOT παρέχει τη δυνατότητα στον καθένα που διαθέτει βασικές γνώσεις προγραμματισμού να μοιραστεί το όραμά της αυτό και να δημιουργήσει τις δικές του εφαρμογές και αυτοματισμούς.

2.2 Γλώσσα προγραμματισμού Java

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρο-συσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους

πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο "πατέρας" της Java, James Gosling, που εργαζόταν εκείνη την εποχή για τη Sun, έκανε ήδη πειραματισμούς πάνω στη C++. Τελικά μετά από λίγο καιρό κατέληξαν με μια πρόταση για το επιτελείο της εταιρίας, η οποία ήταν η γλώσσα Oak. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά) το οποίο ο Gosling είχε έξω από το γραφείο του και έβλεπε κάθε μέρα.

Η Oak ήταν μία γλώσσα που διατηρούσε μεγάλη συγγένεια με τη C++. Παρόλα αυτά είχε πολύ πιο έντονο αντικειμενοστρεφή (object oriented) χαρακτήρα σε σχέση με την C++ και χαρακτηριζόταν για την απλότητα της. Σύντομα οι υπεύθυνοι ανάπτυξης της νέας γλώσσας ανακάλυψαν ότι το όνομα Oak ήταν ήδη κατοχυρωμένο οπότε κατά την διάρκεια μιας εκ των πολλών συναντήσεων σε κάποιο τοπικό καφέ αποφάσισαν να μετονομάσουν το νέο τους δημιούργημα σε Java που εκτός των άλλων ήταν το όνομα αγαπητού καφέ για τους δημιουργούς της. Ο πρώτος μεταγλωττιστής (compiler) της ήταν γραμμένος στη γλώσσα C ενώ το 1994 ξαναγράφεται ο μεταγλωττιστής σε Java. Από εκεί και πέρα η Java ακολουθεί μία ανοδική πορεία και είναι πλέον μία από τις πιο δημοφιλείς γλώσσες στον χώρο της πληροφορικής. Στις 13 Νοεμβρίου του 2006 η Java έγινε πλέον μια γλώσσα ανοιχτού κώδικα (GPL).



Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh και γενικότερα σε οποιαδήποτε πλατφόρμα χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και

εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine).

Αφού γραφεί κάποιο πρόγραμμα σε Java, στη συνέχεια μεταγλωττίζεται μέσω του μεταγλωττιστή javac, ο οποίος παράγει έναν αριθμό από αρχεία .class (κώδικας byte ή bytecode). Ο κώδικας byte είναι η μορφή που παίρνει ο πηγαίος κώδικας της Java όταν μεταγλωττιστεί. Όταν πρόκειται να εκτελεστεί η εφαρμογή σε ένα μηχάνημα, το Java Virtual Machine που πρέπει να είναι εγκατεστημένο σε αυτό θα αναλάβει να διαβάσει τα αρχεία .class. Στη συνέχεια τα μεταφράζει σε γλώσσα μηχανής που να υποστηρίζεται από το λειτουργικό σύστημα και τον επεξεργαστή. Πιο σύγχρονες εφαρμογές της εικονικής Μηχανής μπορούν και μεταγλωττίζουν εκ των προτέρων τμήματα bytecode απευθείας σε κώδικα μηχανής με αποτέλεσμα να βελτιώνεται η ταχύτητα. Χωρίς αυτό δε θα ήταν δυνατή η εκτέλεση λογισμικού γραμμένου σε Java. Πρέπει να σημειωθεί ότι η VM είναι λογισμικό που εξαρτάται από την πλατφόρμα, δηλαδή για κάθε είδος λειτουργικού συστήματος και αρχιτεκτονικής επεξεργαστή υπάρχει διαφορετική έκδοσή του. Έτσι υπάρχουν διαφορετικές JVM για Windows, Linux, Unix, Macintosh, κινητά τηλέφωνα, παιχνιδιομηχανές κλπ.

Οτιδήποτε θέλει να κάνει ο προγραμματιστής (ή ο χρήστης) γίνεται μέσω της εικονικής μηχανής. Αυτό βοηθάει στο να υπάρχει μεγαλύτερη ασφάλεια στο σύστημα γιατί η εικονική μηχανή είναι υπεύθυνη για την επικοινωνία χρήστη - υπολογιστή. Ο προγραμματιστής δεν μπορεί να γράψει κώδικα ο οποίος θα έχει καταστροφικά αποτελέσματα για τον υπολογιστή γιατί η εικονική μηχανή θα τον ανιχνεύσει και δε θα επιτρέψει να εκτελεστεί. Αυτό είναι ιδιαίτερα χρήσιμο για μεγάλα καταναμημένα συστήματα όπου πολλοί χρήστες χρησιμοποιούν το ίδιο πρόγραμμα συγχρόνως.

2.3 Πλατφόρμα Sun SPOT

Η πλατφόρμα Sun SPOT ξεκίνησε το 2003 ως μια μελέτη στις δυνατότητες των ασύρματων τεχνολογιών. Από τότε νέες δυνατότητες ενσωματώνονται ώστε να κάνουν τις μικρές συσκευές μικρότερες, πιο έξυπνες, πιο ασφαλείς και ικανές να ανταπεξέλθουν σε ακόμα μεγαλύτερο πεδίο εφαρμογών. Έχει φτιαχτεί για να ενθαρρύνει δημιουργούς χωρίς καμία προηγούμενη εμπειρία και γνώση

ενσωματωμένων συσκευών να δημιουργήσουν τις δικές τους εφαρμογές και να επεκτείνουν τις δυνατότητες υπαρχόντων συσκευών.



Βασίζοντας τη πλατφόρμα της Sun SPOT πάνω στην τεχνολογία Java, τόσο στο hardware όσο software, έγινε ευκολότερη η συγγραφή κώδικα για μικρές ασύρματες συσκευές αισθητήρων άλλων ηλεκτρονικών συσκευών. Έτσι, οι δημιουργοί μπορούν να γράψουν ένα πρόγραμμα σε Java, να το φορτώσουν σε μια συσκευή να το τρέξουν και να αναζητήσουν τα σφάλματα πάνω στον κώδικα με τα γνωστά Java IDEs. Λόγω της έμφυτης φορητότητας της Java είναι επίσης ευκολότερο να μεταφέρουμε εφαρμογές μεταξύ των διάφορων πλατφόρμων. Οι συσκευές Sun SPOT προσφέρουν μια μικρή, ευέλικτη ασύρματη πλατφόρμα ώστε να τοποθετηθούν αυτές οι εφαρμογές. Για τα εκατομμύρια των χρηστών που χειρίζονται και γράφουν κώδικα σε Java μέσω αυτής της συσκευής μπορούν να επιτύχουν αρκετά πράγματα.

Οι συσκευές που χρησιμοποιούν Java όπως είδαμε πιο πάνω είναι γνωστές για την ικανότητά τους να είναι ανεξάρτητες από το hardware. Το Sun SPOT χρησιμοποιεί ένα μικρό J2ME (Java 2 Platform Micro Edition) το οποίο τρέχει κατευθείαν στον επεξεργαστή χωρίς να απαιτείται λειτουργικό σύστημα αφού όλες οι εφαρμογές απευθύνονται στην Virtual Machine της Java. Μια εντυπωσιακή δυνατότητα είναι η μεταφορά λογισμικού από μια Sun SPOT συσκευή σε μια άλλη μαζί με τα δεδομένα της κατάστασής της ενώ βρίσκονται και οι δύο σε λειτουργία. Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωση που η

μπαταρία μια συσκευής βρίσκεται σε χαμηλό επίπεδο και ολόκληρο το λογισμικό της μεταφέρεται σε μια άλλη με περισσότερη μπαταρία αποφεύγοντας έτσι απώλεια πληροφοριών από την κατάσταση των αισθητήρων.

Αντιθέτως με άλλα συστήματα, οι SUNSPOT χρήστες μπορούν να χρησιμοποιούν τα κλασικά εργαλεία ανάπτυξης εφαρμογών Java. Τα πιο ευρέως διαδεδομένα είναι οι εφαρμογές NetBeans (αναπτύσσεται από την ίδια την Sun) και Eclipse. Αποτελούν ολοκληρωμένες σουίτες λογισμικού για ανάπτυξη κώδικα αλλά και διόρθωση λαθών.

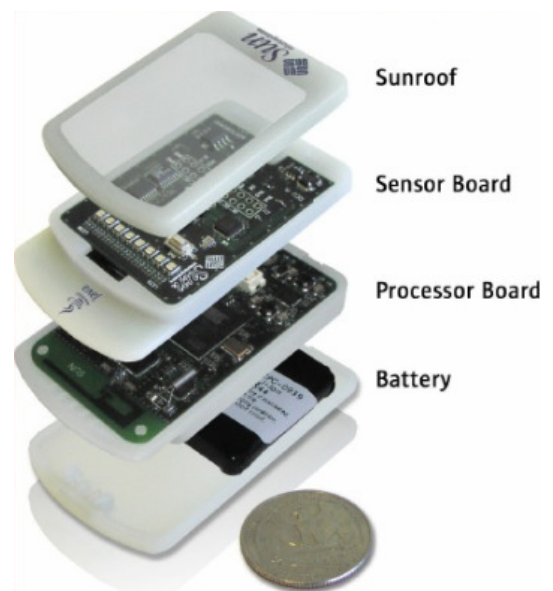
Το J2ME και το SunSPOT είναι ανοικτού κώδικα. Με τον όρο Λογισμικό ανοικτού κώδικα εννοείται λογισμικό του οποίου ο πηγαίος κώδικας διατίθεται ελεύθερα σε αυτούς που θέλουν να τον εξετάσουν, και τροποποιήσουν ή χρησιμοποιήσουν σε άλλες εφαρμογές. Υπάρχουν πολλές διαφορετικές άδειες χρήσης που μπορεί να συνοδεύουν το λογισμικό ανοικτού κώδικα.

2.4 Χαρακτηριστικά κόμβου Sun SPOT

2.4.1 Δομή

Η δομή ενός κόμβου Sun SPOT όπως φαίνεται στο δίπλα σχήμα αποτελείται από 3 στρώματα:

- Τη μπαταρία
- Το ολοκληρωμένο κύκλωμα του επεξεργαστή μαζί με τον ραδιοπομπό (processor board). Αυτό μπορεί να λειτουργήσει και ως κεντρικός σταθμός
- Το ολοκληρωμένο κύκλωμα των αισθητήρων(sensor board)



Το μέγεθος του κόμβου είναι ιδιαίτερα μικρό μεγαλώνοντας το εύρος των εφαρμογών που μπορούν να χρησιμοποιηθούν. Συγκεκριμένα οι διαστάσεις του κόμβου είναι 41 x 23 x 70mm με βάρος 54 gr.

2.4.2 Τεχνικά χαρακτηριστικά

Επεξεργασία και ραδιοπομπός

- CPU 32 bit της οικογένειας επεξεργαστών ARM920T με συχνότητα 180MHz. Είναι αρχιτεκτονικής RISC
- 1Mb RAM μνήμη
- 8Mb μνήμη για αποθήκευση εφαρμογών και δεδομένων Flash
- ο ραδιοπομπός υποστηρίζει το πρότυπο 2.4 GHz IEEE 802.15.4 και διαθέτει ενσωματωμένη κεραία
- ολοκληρωμένο κύκλωμα χρονισμού AT91SAM9G20
- σύνδεση μέσω USB για ανταλλαγή δεδομένων και φόρτιση μπαταρίας

Κύκλωμα αισθητήρων

- επιταχυνσιόμετρο 2G/6G τριών διαστάσεων
- Αισθητήρας θερμοκρασίας
- Light sensor
- 8 tri-color LEDs
- 6 αναλογικές εισόδους
- 2 στιγμιαίους διακόπτες switches
- 5 γενικής χρήσης I/O pins
- 4 υψηλής έντασης ρεύματος output pins

Μπαταρία

- 3.7V επαναφορτιζόμενη μπαταρία lithium-ion των 750 mAh
- Αυτόματη διαχείριση της μπαταρίας από software

Ασφάλεια

Χρησιμοποιώντας τη τεχνολογία Elliptic Curve Cryptography (ECC) έγινε

εφικτό να προστίθεται περισσότερη ασφάλεια χωρίς να χαραμίζεται η μνήμη και η ικανότητα επεξεργασίας των μικρών συσκευών. Η υλοποίηση της Sun του ECC ενεργοποιεί ένα μικρό «αποτύπωμα». Μια εφαρμογή με ονομασία Sizzle το οποίο περιλαμβάνει πρωτόκολλα http και ssl μπορεί να ενσωματωθεί μέσα στις συσκευές ώστε να είναι δυνατή η ασφαλής εποπτεία ο έλεγχος του συστήματος από έναν Web browser

Δίκτυο

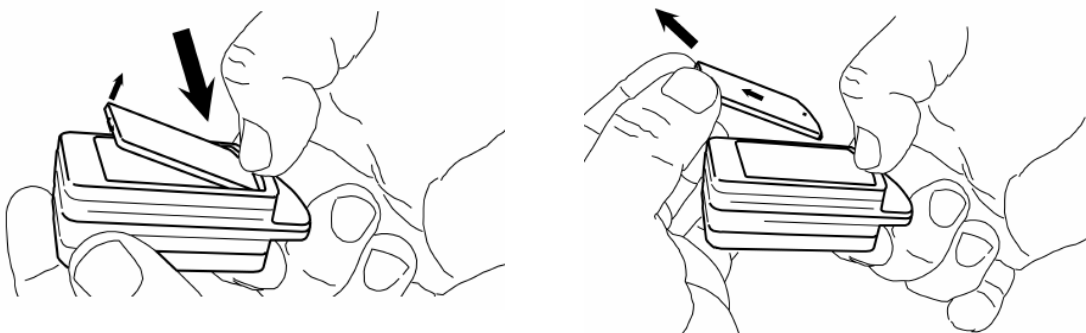
Οι κόμβοι επικοινωνούν χρησιμοποιώντας το πρότυπο IEEE 802.15.4 που περιέχει και το MAC layer, ενώ πάνω από αυτό μπορεί να χρησιμοποιηθεί κάποιο άλλο πρότυπο όπως το Zigbee που περιγράφηκε σε προηγούμενο κεφάλαιο.

Κλιμάκωση

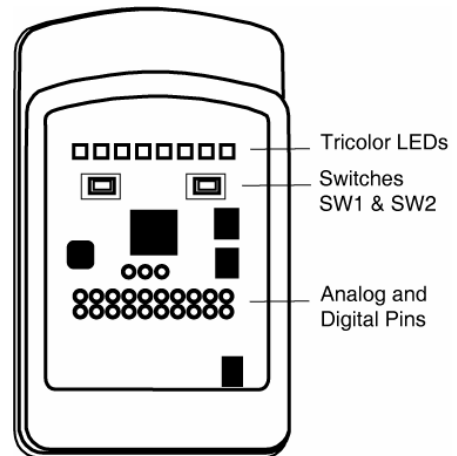
Οποιαδήποτε συσκευή της τεχνολογίας Sun SPOT είναι ικανή να ανταπεξέλθει σε μεγάλη κλιμάκωση πχ. ολοκληρωμένα συστήματα από έναν έως χιλιάδες επεξεργαστές.

2.5 Χρήση Sun SPOT

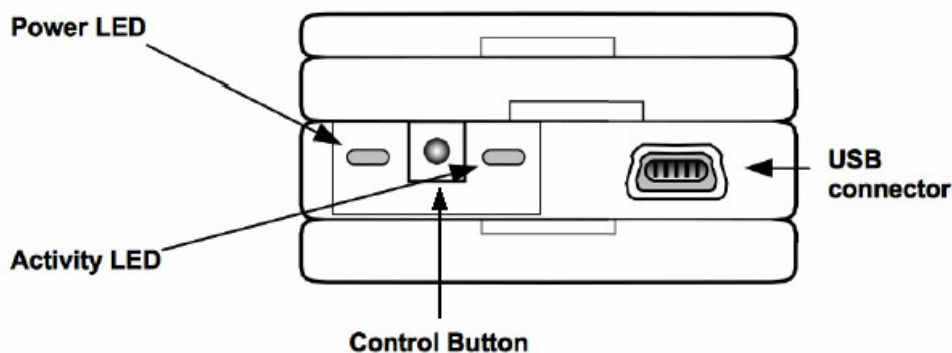
Για να χρησιμοποιήσουμε τα LEDs, τους διακόπτες και τις εισόδους και εξόδους της κάθε μονάδας SunSPOT πρέπει να ανοιχτεί το καπάκι της. Αυτό γίνεται πιέζοντας το καπάκι στην άκρη του απαλά με διεύθυνση προς τα κάτω και πίσω.



Ανοίγοντας το καπάκι έχουμε πρόσβαση στην μονάδα. Όλα μπορούν να ελεγχθούν μέσω προγραμμάτων Java καθορίζοντας συγκεκριμένη συμπεριφορά για το καθένα ανάλογα με την εφαρμογή που τρέχει την κάθε στιγμή. Στο πάνω μέρος της μονάδας βρίσκεται η κεραία για την ασύρματη επικοινωνία με άλλες μονάδες και με τον κεντρικό σταθμό.



Στο πλαϊνό μέρος βρίσκονται η θύρα USB, ο διακόπτης ελέγχου και 2 LEDs, ένα που χαρακτηρίζει την δραστηριότητα (Activity LED) και ένα για το επίπεδο ενέργειας (Power LED). Ο διακόπτης χρησιμοποιείται για να θέσει σε λειτουργία τη μονάδα να την κάνει επανεκκίνηση, αλλά και να την απενεργοποιήσει. Όταν η μονάδα είναι κλειστή, πατώντας απαλά τον διακόπτη τίθεται σε λειτουργία ενώ αναβοσβήνει πράσινο το Activity LED για 2 δευτερόλεπτα. Ενώ βρίσκεται σε λειτουργία, πατώντας πάλι τον διακόπτη γίνεται reset στη μονάδα τερματίζοντας το όποιο πρόγραμμα τρέχει εκείνη τη στιγμή και ξαναφορτώνοντάς το. Πάλι το Activity LED αναβοσβήνει πράσινο για 2 δευτερόλεπτα. Τέλος για να σβήσουμε την μονάδα κρατάμε πατημένο τον διακόπτη για περίπου 5 δευτερόλεπτα. Το Activity LED αναβοσβήνει κόκκινο 2 φορές.



Το Activity LED χρησιμεύει για να δείχνει την κατάσταση επικοινωνίας της μονάδας. Ανάλογα με το χρώμα και την φωτεινότητά του κωδικοποιούνται διαφορετικές καταστάσεις. Συγκεκριμένα:

Activity LED	Σημασία
Αναβοσβήνει γρήγορα πράσινο	Αναζήτηση κεντρικού σταθμού μέσω USB
Αναβοσβήνει πράσινο	Λαμβάνονται πακέτα από τον κεντρικό σταθμό
Αναβοσβήνει κόκκινο	Αποστέλλονται πακέτα προς τον κεντρικό σταθμό
Αναβοσβήνει πράσινο 2 φορές κάθε 12 δευτερόλεπτα	Λειτουργεί ως κεντρικός σταθμός

Το Power LED χρησιμοποιείται για να δηλώσει στον χρήστη το επίπεδο της μπαταρίας. Ομοίως με παραπάνω έχουμε:

Power LED	Σημασία
3 φορές ανάβει κόκκινο	Τερματισμός λειτουργίας
Αργή εναλλαγή έντονου – αχνού πράσινου	Φόρτιση μέσω USB, CPU σε λειτουργία
Αργή εναλλαγή αχνού πράσινου και σβηστού	Φόρτιση μέσω USB, CPU όχι σε λειτουργία
Σταθερά αχνό πράσινο	Συνδεδεμένο σε USB, πλήρως φορτισμένο, CPU σε λειτουργία
Σταθερά αχνό κόκκινο	Χαμηλή μπαταρία
1 φορά αναβοσβήνει πράσινο	Επανεκκίνηση
Ανάβει πράσινο και σταδιακά σβήνει	Έναρξη λειτουργίας

Η συμπεριφορά και των 2 LEDs μπορεί να αλλάξει μέσω του προγράμματος που τρέχει η μονάδα. Αυτή είναι η εργοστασιακή συμπεριφορά τους

2.6 Αγορά Sun SPOT

Ο χρήστης που θέλει να αγοράσει την πλατφόρμα μπορεί να το κάνει μέσα από την επίσημη ιστοσελίδα

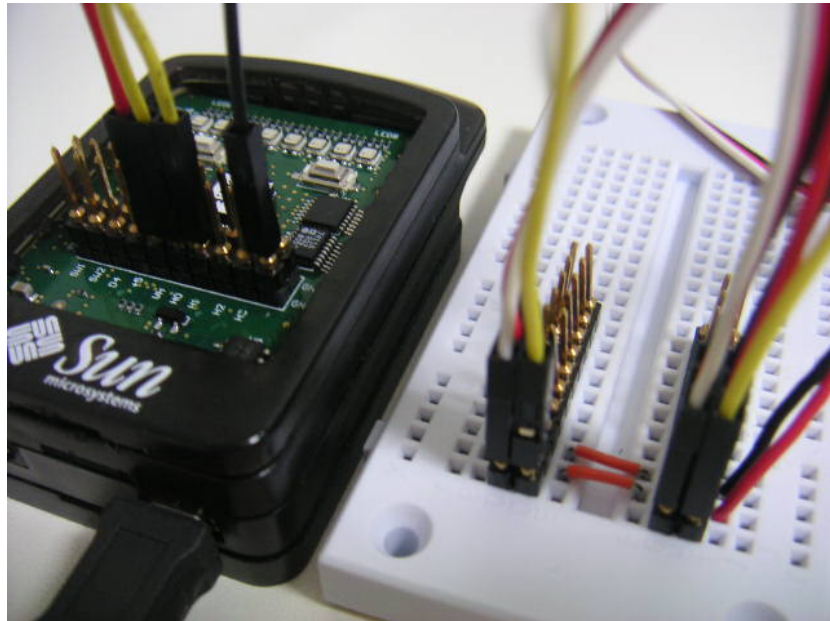
<http://www.sunspotworld.com>.

Το πακέτο που ονομάζεται Sun SPOT Java Development Kit περιλαμβάνει

- 2 Sun SPOT συσκευές
- 1 κεντρικό σταθμό
- 1 usb καλώδιο
- 2 βάσεις τοποθέτησης για τοίχο, μια για κάθε μονάδα
- 1 βάση τοποθέτησης μιας μονάδας πάνω σε πλακέτα ολοκληρωμένου κυκλώματος
- CD με το απαραίτητο λογισμικό για ανάπτυξη εφαρμογών καθώς και εγχειρίδια χρήσης.
- Η τιμή του είναι 399\$



Κεφάλαιο 3



Εφαρμογή με Sun SPOT

3.1 Περιγραφή της εφαρμογής

Η εφαρμογή που υλοποιείται στην παρούσα εργασία αποτελεί παραλλαγή του παραδείγματος (demo) με όνομα *SendDataDemo*, που υπάρχει στο CD του Sun SPOT Java Development Kit και διανέμεται μέσω διαδικτύου από την εταιρία Sun. Ένας αριθμός μονάδων SunSPOT (hosts) διασκορπισμένα στο χώρο παρακολουθούν 2 φυσικά μεγέθη, την ένταση του φωτός και την θερμοκρασία μέσω των αντίστοιχων αισθητήρων τους. Ανά περιοδικά χρονικά διαστήματα λαμβάνουν μετρήσεις τις οποίες και αποστέλλουν στον κεντρικό σταθμό (basestation). Ο τελευταίος λαμβάνει τις μετρήσεις από όλους τους κόμβους, και για κάθε κόμβο αποθηκεύει τις 100 τελευταίες μετρήσεις για κάθε φυσικό μέγεθος. Στη συνέχεια υπολογίζει τις μέσες τιμές θερμοκρασίας και φωτεινότητας για κάθε αισθητήρα. Αν οι τρέχουσες τιμές διαφέρουν πολύ από τις μέσες τότε θα ανιχνεύεται πιθανή αλλαγή στο εποπτευόμενο χώρο.

Η υλοποίηση και δοκιμή της εφαρμογής έγινε στο περιβάλλον προσομοίωσης Solaris που επίσης συμπεριλαμβάνεται στο CD ή μπορεί να το βρει κανείς στο διαδικτυακό τόπο <http://www.sunspotworld.com/>

3.2 Εγκατάσταση απαραίτητων εφαρμογών

Το κύριο πρόγραμμα που χρησιμοποιείται για έλεγχο των μονάδων Sun SPOTs, δημιουργία μιας εφαρμογής αλλά και προσομοίωση είναι το Sun SPOT Manager Tool. Επειδή αυτό είναι γραμμένο σε Java, για να εκτελεστεί απαραίτητη προϋπόθεση είναι να υπάρχει εγκατεστημένη στο σύστημα μια έκδοση Java, δηλαδή το JRE (Java Runtime Environment). Το JRE περιλαμβάνει την Java Virtual Machine η οποία τρέχει όλες τις εφαρμογές. Συνιστώμενη έκδοση είναι η 6 update 21 ή μεγαλύτερη. Το JRE διατίθεται δωρεάν από την εταιρία Oracle (έχει εξαγοράσει την Sun, κατασκευάστρια εταιρία της Java και των SunSPOTS), από την ιστοσελίδα της. Υπάρχουν διαφορετικά JRE ανάλογα με το λειτουργικό σύστημα που πρόκειται να εγκατασταθεί.

Όταν γίνεται για πρώτη φορά εκτέλεση του Sun SPOT Manager Tool, αυτό ελέγχει αν υπάρχουν οι απαραίτητες εφαρμογές για να λειτουργήσει σωστά. Αρχικά ελέγχεται αν υπάρχει το JDK (Java Developer Kit) το οποίο είναι απαραίτητο για συγγραφή και ανάπτυξη εφαρμογών σε Java. Όπως και με το

JRE, μπορεί κάποιος να το βρει από το site της Oracle και υπάρχουν διαφορετικές εκδόσεις ανάλογα με το λειτουργικό σύστημα του υπολογιστή. Μέσα στο JDK περιλαμβάνεται και το JRE. Αν δεν ανιχνευθεί JDK, το Sun SPOT Manager Tool αναλαμβάνει να το κατεβάσει από το internet και να εγκαταστήσει.

Στην συνέχεια ελέγχει αν υπάρχει το Apache Ant. Πρόκειται για ένα πρόγραμμα αυτόματου κατασκευαστή, γραμμένο σε Java, που διαθέτει δικές του βιβλιοθήκες και γραμμή εντολών (command line). Είναι open source και μπορεί, χρησιμοποιώντας το JDK, να κάνει compile και build κώδικα Java χρησιμοποιώντας οδηγίες που βρίσκονται σε xml αρχείο με όνομα build.xml. Μπορεί να το κατεβάσει καθένας δωρεάν από τον ιστότοπο <http://ant.apache.org/> όπου θα βρει και οδηγίες για την εγκατάσταση. Αν δεν υπάρχει στο σύστημα αναλαμβάνει πάλι το Sun SPOT Manager Tool να το κατεβάσει και να το εγκαταστήσει αυτόματα.

Τέλος γίνεται έλεγχος αν υπάρχει εγκατεστημένο το Netbeans. Είναι κατασκευασμένο από την Oracle και πρόκειται για IDE (Integrated Development Environment) δηλαδή για γραφικό περιβάλλον με σκοπό τη διευκόλυνση του προγραμματιστή στην συγγραφή και δημιουργία εφαρμογών σε Java. Περιλαμβάνει πληθώρα επιλογών όπως αυτόματη επισήμανση λαθών κατά τη συγγραφή κώδικα, pop-up menu με τις μεθόδους της κάθε κλάσης, εύκολη δημιουργία κλάσεων και μεθόδων καθώς και compile, build και debugging με ενσωματωμένη γραμμή εντολών. Επίσης είναι δυνατή η εγκατάσταση πρόσθετων λειτουργιών (plugins) για εξειδικευμένες εφαρμογές και εξατομίκευση. Υπάρχει plugin για το SunSPOT για εύκολη συγγραφή κώδικα ανάλογα με τον αν προορίζεται για τα hosts ή για το basestation.

Έχοντας ολοκληρώσει τους ελέγχους και τις εγκαταστάσεις των απαραίτητων εφαρμογών, το Manager Tool εγκαθιστά το SunSPOT SDK που είναι απαραίτητο για τη δημιουργία εφαρμογών και τη φόρτωσή τους στις μονάδες. Επίσης παρέχει τις κατάλληλες εντολές για έλεγχο τόσο των μονάδων όσο και του κεντρικού σταθμού. Περιλαμβάνει το πρόγραμμα Solarium απαραίτητο για την προσομοίωση αλλά και τον έλεγχο των μονάδων.

Τόσο το JDK όσο και το Ant είναι απαραίτητα για την λειτουργία του Sun SPOT Manager Tool, η οποία δεν είναι δυνατή χωρίς αυτά ενώ το Netbeans είναι προαιρετικό. Ο κάθε προγραμματιστής μπορεί να χρησιμοποιήσει

οποιοδήποτε πρόγραμμα για συγγραφή κώδικα, αλλά η συμβατότητα του Netbeans με το Manager Tool το καθιστά ιδανικό. Είναι δυνατή η φόρτωση του Java προγράμματος από το Netbeans απευθείας στη μονάδα SunSpot εφόσον υπάρχει το plugin.

3.3 Χρησιμοποιώντας το Sun SPOT Manager Tool

Έχοντας ολοκληρώσει το στάδιο της εγκατάστασης ανοίγει το SPOT Manager Tool. Στο πάνω μέρος του περιλαμβάνει 7 κατηγορίες (tabs):

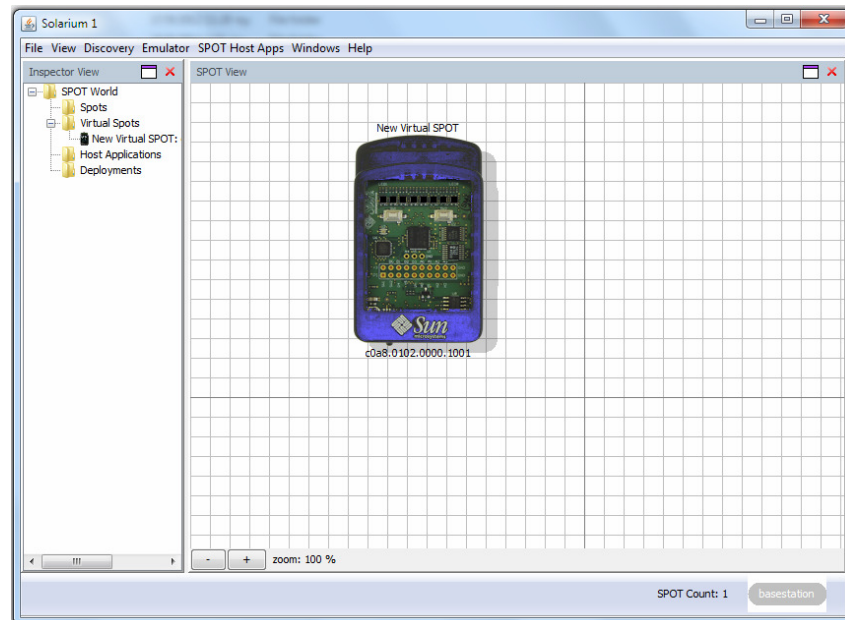


- Sun Spots – Μας δίνει τη δυνατότητα να δούμε και να ελέγξουμε κάθε μονάδα χωριστά. Μπορούμε να τις ενεργοποιήσουμε ή να τις απενεργοποιήσουμε, να πάρουμε πληροφορίες, να φορτώσουμε κάποιο πρόγραμμα ή νέο firmware, ή ακόμα να μετατρέψουμε κάποια από αυτές σε basestation.
- SDKs – Εδώ μπορούμε να εγκαταστήσουμε από το site της Oracle ή να διαγράψουμε εκδόσεις του Sun SPOT SDK.
- Solarium – Από εδώ μπορούμε να εκτελέσουμε το πρόγραμμα Solarium. Με

αυτό διαχειριζόμαστε ομάδες Sun SPOTs καθώς και το πρόγραμμα που τρέχουν. Είναι δυνατή η παρακολούθηση των μετρήσεων κάθε μονάδας. Επίσης μπορεί να γίνει προσομοίωση μιας εφαρμογής με εικονικές μονάδες.

- Docs – Πρόκειται για λίστα από κείμενα με οδηγούς εκμάθησης, εγχειρίδια λειτουργίας, τεχνικά χαρακτηριστικά, APIs (Application Programming Interface) δηλαδή βιβλιοθήκες με κλάσεις Java, κτλ
- Console – Εμφανίζονται μηνύματα διαδικασιών που εκτελούνται από το ίδιο το Manager Tool.
- Share - Δίνει την δυνατότητα αναζήτησης, ανάγνωσης και βαθμολόγησης κώδικα άλλων χρηστών. Ο καθένας μπορεί να μοιραστεί το πρόγραμμά του με άλλους δημιουργούς.
- Preferences – Εδώ ορίζονται επιλογές που επηρεάζουν όλο το πακέτο εφαρμογών. Διαγραφή όλων των SDKs, επιλογές internet και δικτύου κτλ.

3.4 Χρησιμοποιώντας το Solarium

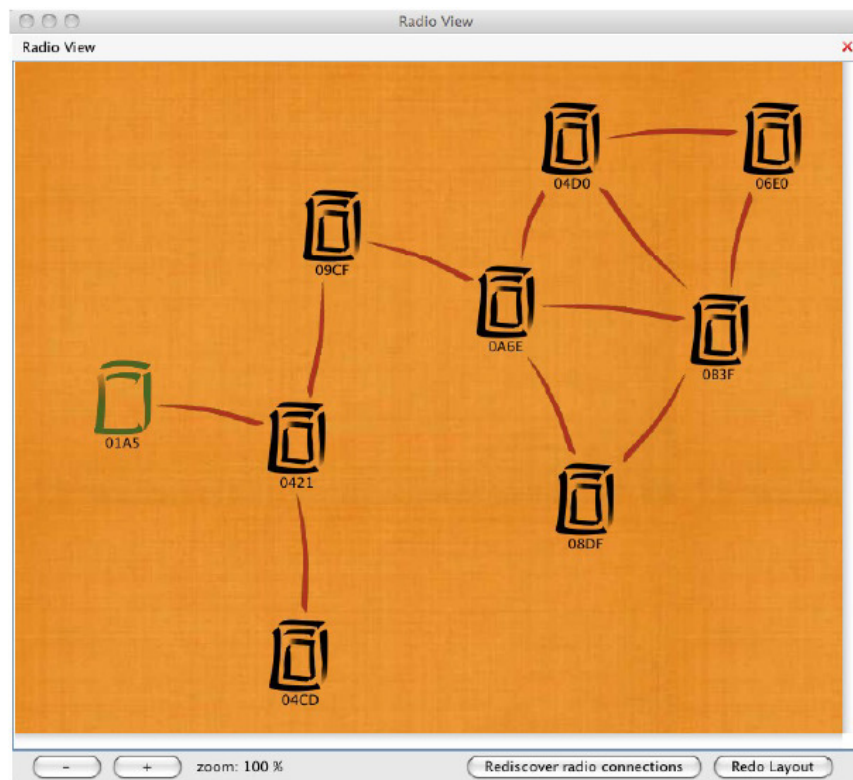


Με το πρόγραμμα αυτό, γραμμένο σε Java, μπορούμε να διαχειριστούμε από απόσταση το δίκτυο των Sun SPOTs. Εντοπίζει αυτόματα τις μονάδες που είναι συνδεδεμένες είτε ασύρματα είτε μέσω USB και τις παρουσιάζει γραφικά.

Μπορεί ο χρήστης να ορίσει αν θα εμφανίζονται όλες οι μονάδες ή μερικές. Επίσης ορίζονται παράμετροι του δικτύου όπως ο αριθμός των hops δηλαδή του αριθμού των αναμεταδόσεων ενός πακέτου του αρχικού SPOT από γειτονικά του μέχρι αυτό να φτάσει στον κεντρικό σταθμό. Επιλέγεται το κανάλι επικοινωνίας και ο χρόνος που περιμένει το πρόγραμμα ώστε να του απαντήσουν οι μονάδες για να τις εντοπίσει.

Μέσω της γραφικής απεικόνισης των SPOTs μπορούμε να ξεκινήσουμε ή να σταματήσουμε τη λειτουργία τους (run MIDlet). Επιλέγοντας τη συγκεκριμένη μονάδα που θέλουμε φορτώνουμε το πρόγραμμα που θέλουμε να τρέξει (Deploy MIDlet bundle). Στο υπόλοιπο menu που εμφανίζεται κάνοντας δεξί κλικ πάνω σε μία μονάδα ξεχωρίζουμε τις επιλογές get info και blink LEDs. Η πρώτη μας δίνει χρήσιμες πληροφορίες για το επίπεδο της μπαταρίας αλλά και για την ενέργεια που καταναλώνει. Η δεύτερη αναβοσβήνει τα LEDs της μονάδας, ιδιαίτερα χρήσιμο για να ανιχνευτεί το συγκεκριμένο SPOT ανάμεσα σε άλλα.

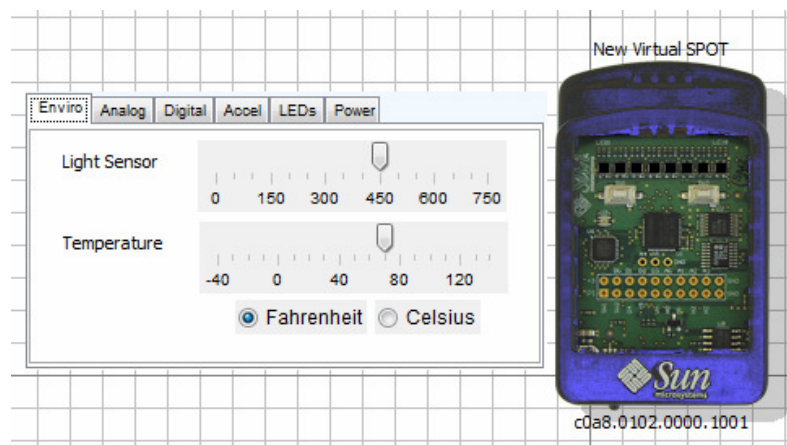
Μέσω του Solarium δίνεται η δυνατότητα απεικόνισης της τοπολογίας του δικτύου αισθητήρων. Προσδιορίζεται γραφικά ο δρόμος που ακολουθούν τα πακέτα κάθε SPOT μέσα στο δίκτυο για να φτάσουν στον κεντρικό σταθμό.



Η λειτουργία αυτή λέγεται Radio View. Όταν εκτελείται ζητείται από το κάθε SPOT να προσδιορίσει τους γείτονές του. Με κόκκινες γραμμές εμφανίζονται ποιες μονάδες μπορούν να ανταλλάξουν πακέτα, ενώ στη συνέχεια μέσω αλγορίθμου υπολογίζεται η καλύτερη δυνατή διαδρομή. Το Radio View δεν γνωρίζει τις πραγματικές θέσεις κάθε μονάδας αλλά μόνο τη θέση της στο δίκτυο.

Παρέχεται στο δημιουργό η επιλογή Deployment View. Σε αυτή εμφανίζονται όλες οι μονάδες στο χώρο. Μπορούμε να εισάγουμε αρχείο με χάρτη και πάνω σε αυτό να καθορίσουμε τις ακριβείς θέσεις των SPOTs. Ελέγχουμε με τον τρόπο αυτό το σύνολο του δικτύου.

Μια από τις πιο σημαντικές λειτουργίες είναι η δυνατότητα της προσομοίωσης. Με τον τρόπο αυτό μπορεί να αναπτύξει εφαρμογές και να τις δοκιμάσει χωρίς να έχει φυσική πρόσβαση στον Sun SPOT Kit. Εισάγουμε εικονικά SPOTs από το menu emulator → new virtual SPOT. Κάθε μονάδα εμφανίζεται στην οθόνη με την δική της εικονική διεύθυνση δικτύου. Πατώντας την επιλογή Display sensor panel εμφανίζεται ένα παράθυρο με όλους τους αισθητήρες της μονάδας. Με τα χειριστήρια μεταβάλλουμε τις τιμές που αυτοί ανιχνεύουν προκαλώντας τις όποιες επιθυμητές αλλαγές. Φορτώνουμε προγράμματα στα εικονικά SPOT με τον ίδιο τρόπο όπως και στα πραγματικά.



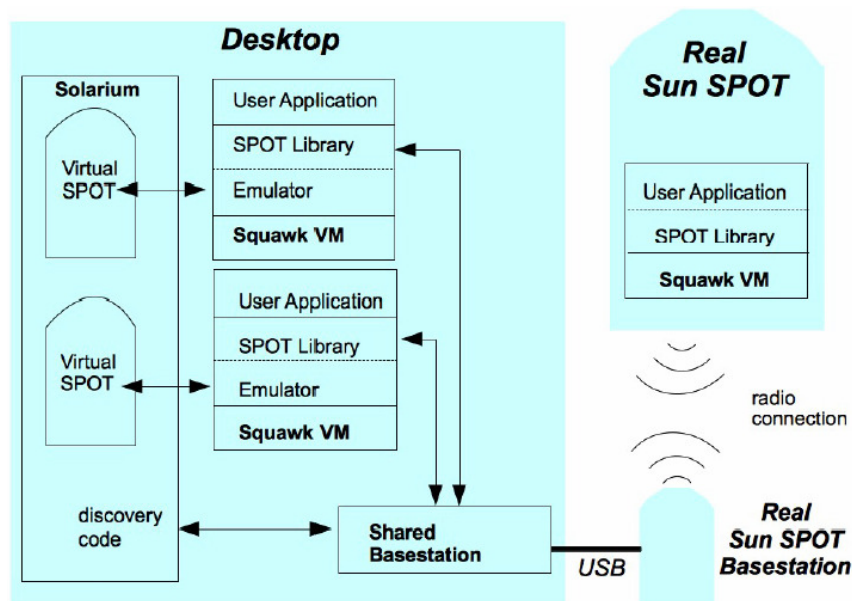
Η προσομοίωση μπορεί να αποτελείται και από πραγματικά και εικονικά SPOTs. Είναι δυνατόν να αντικαταστήσουμε μερικές πραγματικές μονάδες με τις εικονικές τους και η εφαρμογή να τρέχει με συνδυασμό πραγματικών-εικονικών.

Για να ολοκληρωθεί η προσομοίωση δεν είναι αρκετά τα εικονικά μόνο SPOTs. Πρέπει να γίνει προσομοίωση και του basestation. Αυτό γίνεται με τη βοήθεια του Apache Ant. Σε λειτουργικό σύστημα Windows ανοίγουμε το command prompt από το menu Start → Run γράφουμε την εντολή cmd και πατάμε enter. Στο παράθυρο που ανοίγει με dos shell (cd <path>)πηγαίνουμε στο φάκελο που είναι το πρόγραμμα που θα εκτελεί το basestation. Εκεί δίνουμε την εντολή

```
ant host-run -Dbasestation.shared=true -Dbasestation.not.required=true
```

και ξεκινάει η εκτέλεση του προγράμματος του κεντρικού σταθμού. Για σωστή λειτουργία θα πρέπει να έχουμε φτιάξει τα εικονικά SPOTs στο Solarium, να φορτώσουμε το πρόγραμμα που θέλουμε να τρέχουν οι μονάδες, να το τρέξουμε και μετά να τρέξουμε την προσομοίωση της κεντρικής βάσης. Με τον τρόπο αυτό όταν ξεκινήσει η βάση θα τρέχουν ήδη τα SPOTs και θα είναι άμεσα ορατά σε αυτή.

Μέσω του ant μπορούμε να δώσουμε διάφορες παραμέτρους για την προσομοίωσή μας. Η πιο σημαντική είναι η Dbasestation.shared. Αυτή καθορίζει αν τα SPOTs θα επικοινωνούν μεταξύ τους με συνδέσεις point-to-point ή με broadcast. Όταν είναι συνδεδεμένο το basestation στον υπολογιστή και έχει οριστεί shared μπορούν τα εικονικά SPOTs να επικοινωνούν με τα πραγματικά χρησιμοποιώντας τον πομποδέκτη της βάσης.



3.5 Κώδικας της εφαρμογής

Η συγκεκριμένη εφαρμογή όπως περιγράφηκε στην αρχή του κεφαλαίου, αποτελείται από 2 προγράμματα. Το πρόγραμμα που τρέχει στα Sun SPOTs, και παίρνουν τις μετρήσεις και από το πρόγραμμα που τρέχει στη κεντρική βάση (basestation) το οποίο συλλέγει τις μετρήσεις και κάνει τους υπολογισμούς. Οι υπολογισμοί και η αποθήκευση θα μπορούσαν να γίνουν και σε κάθε SPOT καθώς διαθέτει τις δυνατότητες. Αν και προγραμματιστικά αυτό είναι πιο εύκολο σχεδιαστικά είναι λάθος γιατί το πρόγραμμα που τρέχουν οι κόμβοι αισθητήρων πρέπει να είναι μικρό και απλό για εξοικονόμηση πολύτιμης ενέργειας. Παρουσιάζεται ο κώδικας του SPOT:

```
package org.sunspotworld.demo;
import com.sun.spot.io.j2me.radiogram.*;
import com.sun.spot.resources.Resources;
import com.sun.spot.resources.transducers.ITriColorLED;
import com.sun.spot.resources.transducers.ILightSensor;
import com.sun.spot.resources.transducers.ITemperatureInput;
import com.sun.spot.util.Utils;
import javax.microedition.io.*;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

public class SensorSampler extends MIDlet {
    private static final int HOST_PORT = 67;
    private static final int SAMPLE_PERIOD = 500; // in milliseconds
    protected void startApp() throws MIDletStateChangeException {
        RadiogramConnection rCon = null;
        Datagram dg = null;
        String ourAddress = System.getProperty("IEEE_ADDRESS");
        ILightSensor lightSensor = (ILightSensor)Resources.lookup(ILightSensor.class);
        ITemperatureInput tempSensor =
(ITemperatureInput)Resources.lookup(ITemperatureInput.class);
        ITriColorLED led = (ITriColorLED)Resources.lookup(ITriColorLED.class, "LED7");

        System.out.println("Starting sensor sampler application on " + ourAddress + " ...");

        new com.sun.spot.service.BootloaderListenerService().getInstance().start();

        try {
            rCon = (RadiogramConnection) Connector.open("radiogram://broadcast:" +
HOST_PORT);
            dg = rCon.newDatagram(100); // only sending 12 bytes of data
        } catch (Exception e) {
            System.err.println("Caught " + e + " in connection initialization.");
        }
    }
}
```

```

        notifyDestroyed();
    }

    while (true) {
        try {
            long now = System.currentTimeMillis();
            int lightdata = lightSensor.getValue();
            double tempdata = tempSensor.getCelsius();

            led.setRGB(255, 255, 255);
            led.setOn();
            Utils.sleep(50);
            led.setOff();

            dg.reset();
            dg.writeLong(now);
            dg.writeInt(lightdata);
            dg.writeDouble(tempdata);
            rCon.send(dg);

            System.out.println("Light value = " + lightdata);
            System.out.println("Temperature value = " + tempdata);

            Utils.sleep(SAMPLE_PERIOD - (System.currentTimeMillis() - now));
        } catch (Exception e) {
            System.err.println("Caught " + e + " while collecting/sending sensor sample.");
        }
    }
}

protected void pauseApp() {
    // This will never be called by the Squawk VM
}

protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
    // Only called if startApp throws any exception other than MIDletStateChangeException
}
}

```

Η κλάση του SunSpot λέγεται `SensorSampler` και επεκτείνει την κλάση `MIDlet` που χαρακτηρίζει όλες τις εφαρμογές για τους κόμβους. Στο πρώτο τμήμα γίνονται `import` τα κατάλληλα πακέτα του SunSPOT SDK αλλά και του JDK ώστε να μπορούμε να χρησιμοποιήσουμε συγκεκριμένες κλάσεις. Το πακέτο `radiogram` είναι απαραίτητο στη δημιουργία πρωτοκόλλου ασύρματης επικοινωνίας μεταξύ των SPOTs και της βάσης. Οι κλάσεις `ITriColorLED`, `ILightSensor`, `ITemperatureInput`, δημιουργούν αντικείμενα για τα LEDs, τον

αισθητήρα φωτεινότητας και θερμοκρασίας αντίστοιχα. Διαθέτουν κατάλληλες μεθόδους ώστε να μπορούμε να μεταβάλλουμε την κατάστασή τους και να διαβάσουμε τις τιμές τους. Οι κλάσεις του πακέτου `microedition.midlet` χρειάζονται στην ανταλλαγή πληροφοριών μεταξύ των κόμβων. `HOST_PORT` είναι το port στο οποίο συνδέονται τα Spots με την κεντρική βάση και ορίζεται το 67. `SAMPLE_PERIOD` είναι ο χρόνος μεταξύ διαδοχικών δειγματοληψιών από τον αισθητήρα. Ο κόμβος δεν παρακολουθεί συνέχεια τα φυσικά μεγέθη αλλά ανά τακτά χρονικά διαστήματα για εξοικονόμηση ενέργειας. Τότε στέλνει και τις μετρήσεις του στο basestation. Η μέθοδος `startApp` είναι η κύρια μέθοδος που τρέχει και κάνει όλη την εργασία. Τα αντικείμενα των κλάσεων `RadiogramConnection` και `Datagram` δημιουργούν την επικοινωνία των Spots και κατασκευάζουν το πακέτο πληροφοριών που στέλνεται κάθε φορά. Το `datagram` διαθέτει μεθόδους για προσθήκη πληροφοριών στο πακέτο για αποστολή αλλά και εξαγωγή πληροφοριών από το πακέτο. Η μεταβλητή `IEEE_ADDRESS` περιέχει τη διεύθυνση του κόμβου στο δίκτυο. Στη συνέχεια κατασκευάζονται τα αντικείμενα των LEDs και αισθητήρων. Σε μπλοκ `try` κατασκευάζονται και τα αντικείμενα `datagram` και `RadioConnection` ώστε να ανιχνευτεί τυχόν σφάλμα κατά τη δημιουργία της επικοινωνίας (exception `e`). Έπειτα ακολουθεί ένα ατέρμονο loop στο οποίο λαμβάνονται οι μετρήσεις και αποστέλλονται. Δημιουργείται μεταβλητή τύπου `long` με την ώρα που έγινε η μέτρηση. Με τη μέθοδο `getValue()` παίρνουμε την τιμή του αισθητήρα φωτός η οποία επιστρέφει ακέραια τιμή. Η τιμή της θερμοκρασίας λαμβάνεται σε βαθμούς celsius με τη μεθοδο `getCelsius()` και είναι τύπου `double`. Γενικά κάθε αισθητήρας επιστρέφει μέσω της αντίστοιχης μεθόδου του διαφορετικού τύπου μεταβλητή. Στη συνέχεια ορίζεται μέσω μεθόδων του αντικειμένου `led`, να ανάβει 1 LED του SPOT για λίγο κάθε φορά που λαμβάνεται μέτρηση. Μετά δημιουργούμε το πακέτο πληροφορίας για μετάδοση. Με την μέθοδο `reset()` του αντικειμένου `datagram` διαγράφεται το όποιο παλιό περιεχόμενό του. Κάθε τιμή που θέλουμε να βάλουμε στο πακέτο έχει τη δική της μέθοδο ανάλογα με τον τύπο της. Καταλαμβάνει διαφορετικό τμήμα του πακέτου, αυστηρά καθορισμένο, ανάλογα με τα bits που χρησιμοποιεί. Έτσι γίνεται ο διαχωρισμός των δεδομένων μέσα στο πακέτο από τον παραλήπτη. Βάζουμε την ώρα της μέτρησης, τη θερμοκρασία και τη φωτεινότητα. Στη συνέχεια το πακέτο στέλνεται με τη μέθοδο `send()` του αντικειμένου `radiogramconnection`. Τέλος ορίζεται το spot να τυπώνει μήνυμα με έξοδο τις τιμές που μέτρησε. Τα μηνύματα αυτά είναι εμφανή από το Solarium επιλέγοντας από το menu του

εικονικού SPOT το Display application output.

Το πρόγραμμα που τρέχει το basestation είναι το εξής:

```
package org.sunspotworld.demo;

import com.sun.spot.io.j2me.radiogram.*;
import com.sun.spot.peripheral.ota.OTACommandServer;
import java.text.DateFormat;
import java.util.Date;
import java.lang.*;
import java.io.*;
import java.util.*;
import javax.microedition.io.*;
import javax.swing.*;
import com.sun.spot.util.IEEEAddress;

public class SendDataDemoHostApplication{
    private static final int HOST_PORT = 67;

    private void run(int n) throws Exception {
        RadiogramConnection rCon;
        Datagram dg;
        DateFormat fmt = DateFormat.getTimeInstance();
        System.out.println("Number of SunSpots: " + n);
        double [][] data= new double[n][2][100];
        int count[]=new int[n];
        double avglight[]=new double[n];
        double avgtemp[]=new double[n];
        boolean ts=false;
        for(int i=0;i<n;i++){
            count[i]=0;
            avglight[i]=0;
            avgtemp[i]=0;
        }

        try {
            rCon = (RadiogramConnection) Connector.open("radiogram://:" + HOST_PORT);
            dg = rCon.newDatagram(rCon.getMaximumLength());
        } catch (Exception e) {
            System.err.println("setUp caught " + e.getMessage());
            throw e;
        }

        while (true) {
            try {
                rCon.receive(dg);
```

```

String addr = dg.getAddress();
String[] addar=addr.split(".");
Integer nn= Integer.valueOf(addar[3]);
int nnn=nn.intValue()-1001;
long time = dg.readLong();

data[nnn][0][count[nnn]]= (double)dg.readInt();
data[nnn][1][count[nnn]]= (double)dg.readDouble();

System.out.println(fmt.format(new Date(time)) + " from: " + addr + " light = " +
data[nnn][0][count[nnn]] + " temperature = " + data[nnn][1][count[nnn]]);

if(count[nnn]==99){
    avglight[nnn]=0;
    avgtemp[nnn]=0;
    for(int i=0;i<count[nnn];i++){
        avglight[nnn]=data[nnn][0][i]/(count[nnn])+avglight[nnn];
        avgtemp[nnn]=data[nnn][1][i]/(count[nnn])+avgtemp[nnn];
    }
    ts=true;
    System.out.println("\t\tAverage lighting from " + addr + " is " + avglight[nnn] + "
and average temperature is " + avgtemp[nnn]);
}
if(Math.abs(data[nnn][0][count[nnn]]-avglight[nnn])>=100 & ts){
    System.out.println("Lighting detected by " + addr + "has changed
dramatically!!");
}
if(Math.abs(data[nnn][1][count[nnn]]-avgtemp[nnn])>=10 & ts){
    System.out.println("Temperature detected by " + addr + " has changed
dramatically!!");
}
count[nnn]++;
if(count[nnn]==100){
    count[nnn]=0;
}
} catch (Exception e) {
    System.err.println("Caught " + e + " while reading sensor samples.");
    throw e;
}
}
}

public static void main(String[] args) throws Exception {
    OTACommandServer.start("SendDataDemo");
    String input=JOptionPane.showInputDialog(null,"Number of SunSPOTS", "Enter the
max number of SunSPOTS",JOptionPane.QUESTION_MESSAGE);
    Integer num=new Integer(input);

```

```

    int n=num.intValue();
    SendDataDemoHostApplication app = new SendDataDemoHostApplication();
    app.run(n);
}
}

```

Η κλάση της βάσης λέγεται `SendDataDemoHostApplication`. Αρχικά γίνονται `import` τα κατάλληλα πακέτα `java` όπως και `priv`. Συμπληρώνονται τα πακέτα `lang` για τους διάφορους μαθηματικούς υπολογισμούς και το `swing` για τη δημιουργία γραφικού παραθύρου για είσοδο πληροφορίας από το χρήστη. Η κλάση αυτή περιλαμβάνει τη μέθοδο `main` για εκτέλεση όλου του προγράμματος. Σε αυτή στην αρχή εγγράφεται η συγκεκριμένη εφαρμογή στον `OTA Command server` και ξεκινάει τη λειτουργία του. Ο `OTA (over-the-air) Command server` είναι ένα πρόγραμμα που τρέχει στα `SPOTs` και «ακούει» εντολές διαχείρισης που μεταδίδονται από εφαρμογές του συνδεδεμένου υπολογιστή, όπως το `Solarium`. Στη συνέχεια εμφανίζεται στο χρήστη ένα γραφικό παράθυρο εισόδου που του ζητά το μέγιστο αριθμό `SPOTs` που χρησιμοποιούνται κάθε φορά. Ο αριθμός αυτός είναι απαραίτητος για να καθορίζει στο πρόγραμμα την διάσταση του πίνακα μετρήσεων. Για κάθε `SPOT` δεσμεύεται πίνακας μετρήσεων διαστάσεων `2X100` ώστε να αποθηκεύονται 100 μετρήσεις για φωτεινότητα και θερμοκρασία. Συνεπώς πρέπει να ξέρουμε κατά την εκκίνηση τον αριθμό των μονάδων. Ακολούθως δημιουργείται συγκεκριμένο αντικείμενο της κλάσης `SendDataDemoHostApplication` και καλείται η μέθοδος του αντικειμένου `run()`. Αυτή κάνει τη συλλογή δεδομένων από τους κόμβους και αποθηκεύει τις μετρήσεις τους. Καλείται με όρισμα έναν ακέραιο αριθμό που είναι ο αριθμός των `SPOTs` που εισάγαμε προηγουμένως.

Στην μέθοδο `run` σημαντικό ρόλο παίζει ο πίνακας `data`. Πρόκειται για τρισδιάστατο πίνακα `nX2X100` όπου `n` ο αριθμός των `SPOTs`. Για κάθε μονάδα δεσμεύεται ο υποπίνακας `2X100` που αναφέρθηκε πιο πάνω. Για να καθορίσουμε σε ποια θέση του πίνακα θα γράψουμε την κάθε μια από τις 100 μετρήσεις ορίζουμε τον πίνακα `count` διαστάσεως `n`. Κάθε κόμβος έχει δηλαδή το δικό του μετρητή που μετράει πόσες μετρήσεις έκανε. Ο `count` μηδενίζεται αρχικά. Στη συνέχεια στο βρόγχο `while` το `basestation` μπαίνει σε ατέρμονο `loop` που κάθε φορά λαμβάνει ένα πακέτο πληροφορίας από κάποιο `SPOT`. Καθένα διαθέτει τη δική του διεύθυνση στο δίκτυο η οποία στην τελευταία τετράδα αριθμών ξεκινάει από το 1001 για το πρώτο `SPOT` και συνεχίζει με τη σειρά για

τα υπόλοιπα. Για να ξεχωρίσουμε ποιο SPOT εξετάζεται κάθε φορά, ώστε να γράψουμε τα δεδομένα του στην κατάλληλη γραμμή του πίνακα data, αφαιρούμε από την τελευταία τετράδα αριθμών της διεύθυνσης, τον αριθμό 1001 που αντιστοιχεί στο πρώτο SPOT. Με τον τρόπο αυτό για το πρώτο SPOT παίρνουμε τη γραμμή 0 του πίνακα data, για το δεύτερο SPOT τη γραμμή 1 κτλ. Επίσης αποθηκεύουμε τη φωτεινότητα στη στήλη 0 του data και την θερμοκρασία στην 1. Με τον τρόπο αυτό κάθε φορά που εκτελείται το loop λαμβάνουμε μετρήσεις από κάποιο SPOT. Οι θέσεις των πληροφοριών στον πίνακα data καθορίζεται ως εξής:

- η πρώτη διάσταση από την διεύθυνση του SPOT
- η δεύτερη διάσταση από την μέτρηση για θερμοκρασία ή φωτεινότητα
- και η τρίτη από τον πίνακα count που καθορίζει πόσες μετρήσεις έγιναν από κάθε SPOT

Ακολουθως υπολογίζεται η μέση φωτεινότητα και θερμοκρασία των προηγούμενων μετρήσεων του υπό εξέταση SPOT. Για να εντοπίσουμε δραστικές αλλαγές στα μεγέθη ορίζουμε μεταβολή 100 μονάδων για τη φωτεινότητα και 10° για τη θερμοκρασία, ανάμεσα στην τρέχουσα τιμή και τη μέση όλων των προηγούμενων μετρήσεων. Αν εντοπιστεί τέτοια αλλαγή τυπώνεται σχετικό μήνυμα ώστε να ειδοποιείται ο χρήστης. Με την ολοκλήρωση των μετρήσεων και υπολογισμών αυξάνεται κατά ένα το περιεχόμενο του count ώστε να δείχνει στη σωστή τιμή όταν θα ξεκινήσει η επόμενη επανάληψη του while. Τέλος γίνεται έλεγχος αν ο count έχει φτάσει την τιμή 100 οπότε και μηδενίζεται.

3.6 Παρατηρήσεις

Περιορισμός στον κώδικα είναι ο αριθμός των SPOTs. Επειδή χρησιμοποιείται η τελευταία τετράδα της διεύθυνσης μπορούν να χρησιμοποιηθούν τόσα Spots μέχρι να τερματίσει η τελευταία τετράδα. Ο αριθμός αυτός όμως είναι αρκετός. Επίσης προσοχή πρέπει να δοθεί στον χρόνο δειγματοληψίας. Πολύ μικρός χρόνος μπορεί να οδηγήσει σε απώλεια πακέτων από SPOTs καθώς δεν προλαβαίνουν να γίνουν όλοι οι υπολογισμοί. Πρακτικά κάτι τέτοιο είναι δύσκολο μιας και οι υπολογισμοί απαιτούν ελάχιστο χρόνο και τα Spots δεν μπορούν να στέλνουν δεδομένα πολύ σύντομα γιατί καταναλώνουν πολύ ενέργεια. Τέλος τα πακέτα δεδομένων από τα SPOTs δεν έρχονται με την σειρά που καθορίζει η διεύθυνση τους. Δηλαδή δεν φτάνει πρώτα από το 1001 μετά από το 1002 κτλ. Αυτό εξαρτάται από την απόστασή τους, τον αριθμό τους, την διάταξή τους και τα hops. Ο κώδικας δεν έχει πρόβλημα να χειριστεί αυτή την περίπτωση.

Βιβλιογραφία

1. <http://www.sunspotworld.com/>
2. <http://www.oracle.com/index.html>
3. <http://www.sunspotworld.com/docs/Yellow/javadoc/index.html>
4. <http://www.oracle.com/technetwork/java/api-141528.html>
5. “Πλήρες εγχειρίδιο της Java 2” – R. Cadenhead, L. Lemay
6. “Αντικειμενοστρεφής προγραμματισμός JAVA” – Κ. Θραμπουλίδης
7. “Sun SPOT programmers manual” – Sun Labs
8. “Ζητήματα κάλυψης σε ασύρματα δίκτυα αισθητήρων” – Α. Αργύρης, Δ. Κατσαρός Πανεπιστήμιο Βόλου
9. “Ασύρματα δίκτυα αισθητήρων: αξιολόγηση απόδοσης και θέματα υλοποίησης” – Π. Δελλαπόρτας, Γ. Λιοδάκης ΤΕΙ Κρήτης
10. “Ασύρματα δίκτυα αισθητήρων για τη μη επεμβατική παρακολούθηση βιοσήματος” – Α. Λοΐζου, Φ. Κωνσταντίνου ΕΜΠ
11. “Προσομοίωση ασυρμάτων αισθητήρων και μετρήσεις μεγεθών σε πραγματικό χρόνο” – Θ. Κοσσυβάκης, Μ. Δήμου, Χ. Λολιος ΤΕΙ Λάρισας
12. “Σχεδιασμός προσομοίωσης και αξιολόγησης πρωτοκόλλων επικοινωνίας για ασύρματα δίκτυα μικροαισθητήρων” – Α. Κιναλης, Σ. Νικολετσέας Πανεπιστήμιο Πατρών
13. “Αποδοτικά πρωτόκολλα συλλογής δεδομένων από κινητά δίκτυα αισθητήρων” – Κ. Αγγελόπουλος, Σ. Νικολετσέας Πανεπιστήμιο Πατρών
14. “Καταμέτρηση και εντοπισμός θέσεων πολλαπλών πηγών από δίκτυα αισθητήρων” – Γ. Ρουμελιώτης Πανεπιστήμιο Πατρών
15. “Ανάπτυξη κρυπτογραφικών αλγορίθμων για ετερογενή ασύρματα δίκτυα αισθητήρων” – Α. Πυργέλης, Π. Σπυράκης Πανεπιστήμιο Πατρών
16. “Σχεδιασμός και ανάπτυξη ενός γενικού περιβάλλοντος για υλοποίηση εφαρμογών σε ασύρματα δίκτυα αισθητήρων” – Γ. Μυλωνάς, Σ. Νικολετσέας Πανεπιστήμιο Πατρών
17. “Συστήματα επεξεργασίας σημάτων και εικόνων: θεωρία, υλοποιήσεις, εφαρμογές” - Χ. Μαυροκεφαλίδης, Κ. Μπερμπερίδης Πανεπιστήμιο Πατρών